Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources

Boris Katz and **Jimmy Lin** and **Sue Felshin**
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139
{boris, jimmylin, sfelshin}@ai.mit.edu

Abstract

Although vast amounts of information are available electronically today, no effective information access mechanism exists to provide humans with convenient information access. A general, open-domain question answering system is a solution to this problem. We propose an architecture for a collaborative question answering system that contains four primary components: an annotations system for storing knowledge, a ternary expression representation of language, a transformational rule system for handling some complexities of language, and a collaborative mechanism by which ordinary users can contribute new knowledge by teaching the system new information. We have developed a initial prototype, called Webnotator, with which to test these ideas.

1 Introduction

A tremendous amount of heterogenous information exists in electronic format (the most prominent example being the World Wide Web), but the potential of this large body of knowledge remains unrealized due to the lack of an effective information access method. Because natural language is the most convenient and most intuitive method of accessing this information, people should be able to access information using a system capable of understanding and answering natural language questions—in short, a system that combines human-level understanding with the infallible memory of a computer.

Natural language processing has had its successes and failures over the past decades; while the successes are significant, computers will not soon be able to *fully* process and understand language. In addition to the traditional difficulties associated with syntactic analysis, there remains many other problems to be solved, e.g., semantic interpretation, ambiguity resolution, discourse modeling, inferencing, common sense, etc. Furthermore, not all information on the Web is textual—some is sound, pictures, video, etc. While natural language processing is advanced enough to understand typical interactive *questions* about knowledge (interactive questions are typically fairly simple in structure), it cannot understand the *knowledge* itself. For the time being, therefore, the only way for computers to access their own knowledge is for humans to tell the computers what the knowledge means in a language that the computers can understand—but still in a language that humans can produce. A good way to accomplish this is with the use of *natural language annotations*, sentences which are simple enough for a computer to analyze, yet which are in natural human language. Once knowledge is so annotated, and indexed in a knowledge repository, a question answering system can retrieve it.

The START (SynTactic Analysis using Reversible Transformations) Natural Language System (Katz, 1990; Katz, 1997) is an example of a question answering system that uses natural language annotations. START is a natural language question answering system that has been available to users on the World Wide Web[1] since December, 1993. During this time, it has engaged in millions of exchanges with hundreds of thousands of people all over the world, supplying users with knowledge regarding geography, weather, movies, corporations, and many many other areas. Despite the success of START in serving real users, its domain of expertise is relatively small and expanding its knowledge base is a time-consuming task that requires trained individuals.

We believe that the popularity of the Web may offer a solution to this knowledge acquisition problem by providing collaborative mechanisms on a scale that has not existed before. We can potentially leverage millions of users on the World Wide

[1] http://www.ai.mit.edu/projects/infolab

Web to construct and annotate a knowledge base for question answering. In fact, we had proposed a distributed mechanism for gathering knowledge from the World Wide Web in 1997 (Katz, 1997), but only recently have we attempted to implement this idea.

An advantage of natural language annotations is that it paves a smooth path of transition as natural language processing technology improves. As natural language analysis techniques advance, the annotations may become more and more complex. Eventually, a textual information segment could be its own annotation; someday, through other technologies such as speech and image recognition, etc., annotations could even be automatically constructed for non-textual information.

A further advantage is that natural language annotations can be processed via techniques that only partially understand them—via IR engines, or less-than-ideal natural language systems—yet they retain their more complex content and can be reanalyzed at a later date by more sophisticated systems.

2 Overview

We propose a collaborative question answering architecture composed of the four following components:

1. **Natural Language Annotation** is a technique of describing the content of information segments in machine parsable natural language sentences and phrases.

2. **Ternary Expressions** are subject-relation-object triples that are expressive enough to represent natural language, and also amenable to rapid, large-scale indexing.

3. **Transformational Rules** handle the problem of *linguistic variation* (the phenomenon in which sentences with different surface structures share the same semantic content) by explicitly equating representational structures (derived from different surface forms) that have approximately the same meaning.

4. **Collaborative Knowledge Gathering** is a technique by which the World Wide Web may be viewed not only as a knowledge resource, but also a human resource. The knowledge base of a question answering system could be constructed by enlisting the help of millions of ordinary users all over the Web.

3 Annotations

Natural language annotations are machine-parsable sentences or phrases that describe the content of various information segments. They describe the questions that a particular segment of information is capable of answering. For example, the following paragraph about polar bears:

> Most polar bears live along the northern coasts of Canada, Greenland, and Russia, and on islands of the Arctic Ocean...

may be annotated with one or more of the following:

> Polar bears live in the Arctic.
> Where do polar bears live?
> habitat of polar bears

A question answering system would parse these annotations and store the parsed structures with pointers back to the original information segment that they described. To answer a question, the user query would be compared against the annotations stored in the knowledge base. Because this match occurs at the level of ternary expressions, structural relations and transformation (to be discussed in Section 5) can equate queries and annotations even if their surface forms were different. Furthermore, linguistically sophisticated machinery such as synonymy/hyponymy, ontologies, can be brought to bear on the matching process. If a match were found, the segment corresponding to the annotation would be returned to the user as the answer.

The annotation mechanism we have outlined serves as a good basis for constructing a question answering system because annotating information segments with natural language is simple and intuitive. The only requirement is that annotations be machine parsable, and thus the sophistication of annotations depends on the parser itself. As natural language understanding technology improves, we can use more and more sophisticated annotations.

In addition, annotations can be written to describe any type of information, e.g., text, images, sound clips, videos, and even multimedia. This allows integration of heterogenous information sources into a single framework.

Due to the vast size of the World Wide Web, trying to catalog all knowledge on the World Wide Web is a daunting task. Instead, focusing on meta-knowledge is a more promising approach to building a knowledge base that spans more than a tiny fraction of the Web. Consider that reference librarians at large libraries obviously don't know all the knowledge stored in the reference books, but they are nevertheless helpful in finding information, precisely because they have a lot of *knowledge about the knowledge*. Natural language annotations can assist in creating a smart "reference librarian" for the World Wide Web.

4 Representing Natural Language

A good representational structure for natural language is ternary expressions.[2] They may be intuitively viewed as subject-relation-object triples, and can express most types of syntactic relations between various entities within a sentence. We believe that the expressiveness of ternary relations is adequate for capturing the information need of users and the meaning of annotations. For example, "What is the population of Zimbabwe?" would be represented as two ternary expressions:

```
[what is population]
[population of Zimbabwe]
```

Ternary expressions can capture many relationships between entities within a sentence. Such a representational structure is better than a keyword-based scheme which equates a document's keyword statistics with its semantic content. Consider the following sets of sentences/phrases that have similar word content, but (dramatically) different meanings:[3]

(1) The bird ate the young snake.
(1′) The snake ate the young bird.
(2) The meaning of life
(2′) A meaningful life
(3) The bank of the river
(3′) The bank near the river

Ternary expressions abstract away the linear order of words in a sentence into a structure that is closer to meaning, and therefore a relations-based information access system will produce much more precise results.

We have conducted some initial information retrieval experiments comparing a keyword-based approach with one that performs matching based on relations[4]. Using Minipar (Lin, 1999), we parsed the entire contents of the Worldbook Encyclopedia and extracted salient relations from it (e.g., subject-verb-object, possessives, prepositional phrase, etc.) We found that precision for relations-based retrieval was much higher than for keyword-based retrieval. In one test, retrieval based on relations returned the database's three correct entries:

Question: What do frogs eat?

Answer:

(R1) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.
(R4) One group of South American frogs feeds mainly on other frogs.
(R6) Frogs eat many other animals, including spiders, flies, and worms.

compared to 33 results containing the keywords *frog* and *eat* which were returned by the keyword-based system—the additional results all answer a different question ("What eats frogs?") or otherwise coincidentally contain those two terms.

Question: What do frogs eat?

Answer:

...
(R7) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.
(R8) Bowfins eat mainly other fish, frogs, and crayfish.
(R9) Most cobras eat many kinds of animals, such as frogs, fishes, birds, and various small mammals.
(R10) One group of South American frogs feeds mainly on other frogs.
(R11) Cranes eat a variety of foods, including frogs, fishes, birds, and various small mammals.
(R12) Frogs eat many other animals, including spiders, flies, and worms.
(R13) ...

Another advantage of ternary expressions is that it becomes easier to write explicit transformational rules that encode specific linguistic variations. These rules are capable of equating structures derived from different sentences with the same meaning (to be discussed in detail later).

In addition to being adequately expressive for our purposes, ternary expressions are also highly amenable to rapid large-scale indexing and retrieval. This is an important quality because a large question answering system could potentially contain answers to millions of questions. Thus, compactness of representation and efficiency of retrieval become an important consideration. Ternary expressions may be indexed and

[2] See (Katz, 1990; Katz, 1997) for details about such representation in START.

[3] Examples taken from (Loper, 2000)

[4] to be published

retrieved efficiently because they may be viewed using a relational model of data and manipulated using relational databases.

5 Handling Linguistic Variation

Linguistic variation is the phenomenon in which the same meaning can be expressed in a variety of different ways. Consider these questions, which ask for exactly the same item of information:

(4) What is the capital of Taiwan?
(5) What's the capital city of Taiwan?
(6) What is Taiwan's capital?

Linguistic variations can occur at all levels of language; the examples above demonstrate lexical, morphological and syntactic variations. Linguistic variations may sometimes be quite complicated, as in the following example, which demonstrates verb argument alternation.[5]

(7) Whose declaration of guilt shocked the country?
(8) Who shocked the country with his declaration of guilt?

Transformational rules provide a mechanism to explicitly equate alternate realizations of the same meaning at the level of ternary expressions.

As an example, Figure 1 shows a sample transformational rule for (7) and (8).[6] Thus, through application of this rule, question (7) can be equated with question (8).

[n_1 shock n_2]		[n_3 shock n_2]
[shock with n_3]	↔	
[n_3 related-to n_1]		[n_3 related-to n_1]
where $n \in$ *Nouns*		where $n \in$ *Nouns*

Figure 1: Sample Transformational Rule

Transformational rules may be generalized by associating arbitrary conditions with them; e.g., *verb* $\in$ shock, surprise, excite ...

A general observation about English verbs is that they divide into "classes," where verbs in the same class undergo the same alternations. For example, the verbs 'shock', 'surprise', 'excite', etc., participate in the alternation shown in Sentence (7) and (8) not by coincidence, but because they share certain semantic qualities. Although the transformational rule required to handle this alternation is very specific (in that it applies to a very specific pattern of ternary expression structure), the rule can nevertheless be generalized over all verbs in the same class by associating with the rule conditions that must be met for the rule to fire, i.e., *verb* $\in$ *emotional-reaction-verbs*; see Figure 2.

[n_1 v_1 n_2]		[n_3 v_1 n_2]
[v_1 with n_3]	↔	
[n_3 related-to n_1]		[n_3 related-to n_1]
where $n \in$ *Nouns* and $v \in$ *emotional-reaction-verbs*		

Figure 2: Sample Transformational Rule

Note that transformational rules can also encode semantic knowledge and even elements of common sense. For example, a rule can be written that equates a *selling* action with a *buying* action (with verb arguments in different positions). Or as another example, rules can even encode implicatures, e.g., A murdered B implies that B is dead.

Transformational rules can apply at the syntactic, semantic, or even pragmatic levels, and offer a convenient, powerful, and expressive framework for handling linguistic variations.

In order for a question answering system to be successful and have adequate linguistic coverage, it must have a large number of these rules. A lexicon which classified verbs by argument alternation patterns would be a good start, but this is another resource lacking in the world today. Rules generally may be quite complex, and it would be difficult to gather such knowledge from average Web users with little linguistic background. Requesting that users describe segments with multiple annotations (each representing a different phrasing of the description), might serve as a preliminary solution to the linguistic variation problem. Another possible solution will involve learning transformational rules from a corpus. The difficulty in creating transformational rules is a serious problem and unless and until this problem is solved, an NL-based QA system would have to be restricted to a limited domain where a small number of experts could provide enough transformational rule coverage, or would require a large commitment of resources to attain sufficient coverage.

6 Collaboration on the Web

A critical component of a successful natural language question answering system is the knowledge

[5] Beth Levin (Levin, 1993) offers an excellent treatment on English verb classes and verb argument alternations.

[6] This rule is bidirectional in the sense that each side of the rule implies the other side. The rule is actually used in only one direction, so that we canonicalize the representation.

base itself. Although the annotation mechanism simplifies the task of building a knowledge base, the accumulation of knowledge is nevertheless a time consuming and labor intensive task. However, due to the simplicity of natural language annotations (i.e., describing knowledge in everyday English), ordinary users with no technical skills may contribute to a knowledge base. Thus, by providing a general framework in which people on the World Wide Web can enter additional knowledge, we can engage millions of potential users all over the world to collaboratively construct a question answering system. We can distribute the effort of building a knowledge base across many ordinary users by allowing them to teach the system new knowledge.

The idea of using the Internet as a tool for collaboration across geographically distributed regions is not a new idea. The Open Source movement first demonstrated the effectiveness and sustainability of programming computer systems in a distributed manner. Made possible in part by the World Wide Web, the Open Source movement promotes software development by nurturing a community of individual contributors working on freely distributed source code. Under this development model, software reliability and quality is ensured through independent peer review by a large number of programmers. Successful Open Source projects include *Linux*, a popular Unix-like operating system; *Apache*, the most popular Web server in the World; *SendMail*, an utility on virtually every Unix machine; and *dmoz*, the Open Directory Project, whose goal is to produce the most comprehensive directory of the Web by relying on volunteer editors.[7]

Another example of Web-based collaboration is the Open Mind Initiative (Stork, 1999; Stork, 2000), which is a recent effort to organize ordinary users on the World Wide Web (*netizens*) to assist in developing intelligent software. Based on the observation that many tasks such as speech recognition and character recognition require vast quantities of training data, the initiative attempts to provide a collaborate framework for collecting data from the World Wide Web. The three primary contributors within such a framework are domain experts, who provide fundamental algorithms, tool/infrastructure developers, who develop the framework for capturing data, and non-expert *netizens*, who supply the raw training data.

Open Mind Commonsense[8] is an attempt at constructing a large common sense database by collecting assertions from users all over the Web.[9]

Other projects have demonstrated the viability of Web-enabled collaborative problem-solving by harnessing the computational power of idle processors connected to the Web.[10] The SETI (Search for Extraterrestrial Intelligence) Institute was founded after NASA canceled its High Resolution Microwave Survey project. The institute organizes thousands of individuals who donate their idle processor cycles to search small segments of radio telescope logs for signs of extraterrestrial intelligence.[11] Other similar projects that organize the usage of idle processor time on personal computers include the Internet Mersenne Prime Search,[12] and the RC5 Challenge.[13]

Recent technical, social, and economic developments have made the abovementioned models of collaboration possible. Furthermore, numerous successful projects have already demonstrated the effectiveness of these collaborative models. Thus, it is time to capitalize on these emerging trends to create the first collaborative question answering system on the World Wide Web.

Even with the components such as those described above, there still remains a major hurdle in jumpstarting the construction of a collaborative question answering system. We are faced with a classic chicken-and-egg problem: in order to attract users to contribute knowledge, the system must serve a real information need (i.e., actually provide users with answers). However, in order to serve user information needs, the system needs knowledge, which must be contributed by users.

In the initial stages of building a question answering system, the knowledge base will be too sparse to be useful. Furthermore, the system may be very brittle, and might not retrieve the correct information segment, even if it did exist within the knowledge base (e.g., due to a missing transformational rule).

It may be possible to address this dilemma with an incremental approach. The system can first be restricted to a very limited domain (e.g., "animals" or "geography"). Users' expectations will be carefully managed so that they realize the system is highly experimental and has a very limited range of knowledge. In effect, the users will

[7] `http://www.dmoz.org`

[8] `http://openmind.media.mit.edu`

[9] A non-collaborative approach to building a common sense knowledge base is taken by Lenat whose Cyc project (Lenat, 1995) is an attempt to build a common sense knowledge base through a small team of dedicated and highly trained specialists.

[10] `http://www.distributed.org`

[11] `http://setiathome.ssl.berkeley.edu`

[12] `http://www.mersenne.org`

[13] `http://www.distributed.org/rc5/`

be populating a domain-specific knowledge base. Over time, the system will be able to answer more and more questions in that domain, and hence begin to offer interesting answers to real users. After this, a critical mass will form so that users are not only teaching the system new knowledge, but also receiving high quality answers to their questions. At that point, a decision can be made to increase the domain coverage of the system.

In order to initialize this process, we can bootstrap off the curiosity and altruism of individual users. As an example, the Openmind Common Sense project has accumulated over 280 thousand items of information by over six thousand users based on a data collection model that does not supply the user with any useful service. The dream of building "smart" systems has always been a fascination in our culture (e.g., HAL from 2001: A Space Odyssey); we believe that this will serve to attract first-time users.

7 Evolving the System

While the collaborative information gathering task proceeds, we are then faced with the problem of maintaining the system and ensuring that it will provide users with useful information. Two immediate issues arise: quality control and linguistic variation.

How can we insure the quality of the contributed material? In general, any system that solicits information from the World Wide Web faces a problem of quality control and moderation. Although most Web users are well-meaning, a small fraction of Web users may have malicious intentions. Therefore, some filtering mechanisms must be implemented to exclude inappropriate content (e.g., pornography or commercial advertisement) from being inserted into the knowledge base. More troublesome is the possibility of well-meant but incorrect information which is probably more common and definitely harder to detect.

How can we handle linguistic variation? There are often different ways of asking the same question; the annotation of a particular segment might not match the user query, and hence the correct answer may not be returned as a result. Transformational rules may be a solution to the problem, but writing and compiling these rules remain a difficult problem.

We propose a variety of solutions for the maintenance of a collaborative question answering system, depending on the level of human intervention and supervision.

At one end of the spectrum, an unsupervised approach to quality control can be implemented through a distributed system of moderation with different *trust levels*. The scheme essentially calls for self-management of the knowledge repository by the users themselves (i.e., the users with high trust levels). Different trust levels will allow users various levels of access to the knowledge base, e.g., the ability to modify or delete information segments and their associated annotations or to modify other users' trust levels. To initiate the process, only a small group of core editors is required.

In such an unsupervised system, the problem of linguistic variation could be addressed by prompting users to give multiple annotations, each describing the information content of a particular segment in a different way. With a sufficiently large user base, wide coverage might still be achieved in the absence of broad-coverage transformational rules.

At the other end of the spectrum, a large organization may commit significant amounts of resources to maintaining a supervised collaborative knowledge base. For example, an organization may be willing to commit resources to preserve its organizational memory in the form of an "intelligent FAQ" supported by natural language annotations. Computers can be effectively utilized to augment the memory of an organization (Allen, 1977), and have been successfully deployed in real-world environments with relative success (Ackerman, 1998).

If an organization were willing to commit significant resources to a collaborative knowledge repository, then transformational rules can be written by experts with linguistic background. Such experts could constantly review the annotations entered by ordinary users and formulate transformational rules to capture generalizations.

Supervised use of natural language annotation falls short of the grandiose goal of accessing the entire World Wide Web, but is the practical and useful way to apply NL annotation until the transformational rule problem can be solved for unlimited domains.

8 Initial Prototype

Webnotator is a prototype test-bed to evaluate the practicality of NL-based annotation and retrieval through Web-based collaboration. It provides efficient facilities for retrieving answers already stored within the knowledge base and a scalable framework for ordinary users to contribute knowledge.

The system analyzes natural language annotations to produce ternary expressions by postprocessing the results of Minipar (Lin, 1993; Lin, 1994), a fast and robust functional dependency parser that is freely available for non-commercial purposes. The quality of the representational structures depends ultimately on the quality of whatever parser Webnotator is made to access. In the current implementation, ternary expressions are not embedded, elements of ternary expressions are not indexed, and coreference is not detected. Words are stemmed to their root form and morphological information is discarded. The system also implements a version of transformational rules described above as a simple forward-chaining rule-based system.

Using a relational database, Webnotator implements a knowledge base that stores ternary expressions derived from annotations and their associated information segments. Ternary expressions fit neatly into a relational model of data, and thus manipulation of the knowledge (including answering queries and inserting new knowledge) can be formulated as SQL queries. This vastly simplifies development efforts while maintaining robustness and performance.

Webnotator provides an interface through which users may teach the system new knowledge by supplying new information segments and adding new annotations. Essentially, the user enters, in a CGI form, an information segment and annotations that describe the knowledge. Since the segment of information can contain any valid HTML, images, tables, and even multimedia content may be included. Alternatively, the user may simply provide a URL to annotate, and Webnotator will automatically create a link to the URL in its knowledge base.

Currently, Webnotator is a prototype that has been released to a small community of developers and testers within the MIT Artificial Intelligence Laboratory. We plan on releasing the system to the general public in the near future. By collecting knowledge from the general public and by varying the representations and transformations applied by Webnotator, it should be possible to discover which features are most important for a natural-language-based annotation system and whether the state of the art is indeed sufficiently advanced to make such a system practical and effective.

9 Related Work

A variety of research has been conducted on better information access methods on the World Wide Web (e.g., the “Semantic Web” (Berners-Lee, 1999)). However, most of these approaches have concentrated on methods of annotating existing web pages with metadata such as XML/RDF (Resource Description Framework) (Staab et al., 2000), extensions to HTML (Luke et al., 1997; Heflin et al., 1999; Staab et al., 2000), specialized descriptions (W. Dalitz and Lugger, 1997), or even conceptual graphs (Martin and Eklund, 1999).

The common thread among previous work is the embedding of metadata directly into Web documents, which are then gathered via crawling or spidering. This approach only works if the target community of the system is well-defined; adoption of various metadata techniques are presently limited, and thus it would be pointless to crawl the entire web to search for metadata. A model in which distributed metadata are gathered by a spider will not work with a constantly changing community that is ill-defined. In principle, there is no reason why our natural language annotations cannot be embedded into Web documents also; the issue is strictly a practical concern.

Another common theme in previous work is the organization of knowledge in accordance with some pre-established ontology. This presents several challenges for building a general system for gathering knowledge. Ontologies are often either too specific to be of general use (e.g., RiboWeb’s ontology for ribosome data (Altmann et al., 1999)), or too weak to provide much structure (e.g., Yahoo). Since the ontology is static and must be agreed upon prior to any knowledge base development, it may be too constricting and too inconvenient for the expression of new or unanticipated concepts. Although systems do allow for arbitrary extension of the ontology (Heflin et al., 1999; Staab et al., 2000), such extensions defeat the purpose of a structure-imposing ontology. Our proposed alternative to a ontological hierarchy is to take advantage of the expressiveness of natural language, and use linguistic devices to relate concepts. The combination of lexical resources (e.g., synonyms and meronyms in WordNet) and transformational rules provide a natural, extensible way to relate and structure different concepts.

A compelling argument for natural language annotations is their expressiveness and compactness. Martin and Eklund (Martin and Eklund, 1999) argue against an XML-based system of metadata because XML was primarily intended to be machine readable, not human readable. In their paper, they started with an English phrase, and then proceeded to demonstrate the encoding of that sentence in various formalisms. A constraint graph encoding was simpler than a KIF (Knowledge Interchange Format) encoding, which was in turn shorter than a RDF format. Of course, this begs the question: why not just annotate the document with the original English phrase? Current NLP technology can handle a large variety of English sentences and phrases, which may serve as the annotations directly. Such is system is not only simpler, more intuitive, but also more compact.

10 Conclusion

Recent social, technical, and economic developments have made possible a new paradigm of software development and problem solving through loosely-organized collaboration of individuals on the World Wide Web. Many successful precedents have already proven the viability of this approach. By leveraging this trend with existing annotation and natural language technology, we can provide a flexible framework for a question answering system that grows and "evolves" as each user contributes to the knowledge base, with only minimal outside supervision. Testing will reveal whether such a system can help users realize some of the untapped potential of the World Wide Web and other sources of digital information as a vast repository of human knowledge.

References

Mark S. Ackerman. 1998. Augmenting organizational memory: A field study of answer garden. *ACM Transactions on Information Systems*, 16(3):203–224, July.

Thomas Allen. 1977. *Managing the Flow of Technology*. MIT Press.

R. Altmann, M. Bada, X. Chai, M. W. Carillo, R. Chen, and N. Abernethy. 1999. RiboWeb: An ontology-based system for collaborative molecular biology. *IEEE Intelligent Systems*, 14(5):68–76.

T. Berners-Lee. 1999. *Weaving the Web*. Harper, New York.

Jeff Heflin, James Hendler, and Sean Luke. 1999. SHOE: A knowledge representation language for internet applications. Technical Report CS-TR-4078, Institute of Advanced Computer Studies, University of Maryland, College Park.

Boris Katz. 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.

Boris Katz. 1997. Annotating the World Wide Web using natural language. In *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)*.

Doug Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

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

Dekang Lin. 1993. Principled-based parsing without overgeneration. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics (ACL'93)*.

Dekang Lin. 1994. PRINCIPAR—An efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING '94)*.

Dekang Lin. 1999. Minipar—a minimalist parser. In *Maryland Linguistics Colloquium*, University of Maryland, College Park, March 12,.

Edward Loper. 2000. Applying semantic relation extraction to information retrieval. Master's thesis, Massachusetts Institute of Technology.

S. Luke, L. Spector, D. Rager, and J. Hendler. 1997. Ontology-based web agents. In *Proceedings of the First International Conference on Autonomous Agents*.

Philippe Martin and Peter Eklund. 1999. Embedding knowledge in web documents. In *Proceedings of the Eighth International World Wide Web Conference*.

S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. 2000. Semantic community web portals. In *Proceedings of the Ninth International World Wide Web Conference*.

David G. Stork. 1999. Character and document research in the open mind initiative. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*.

David G. Stork. 2000. Open data collection for training intelligent software in the open mind initiative. In *Proceedings of the Engineering Intelligent Systems Symposium (EIS '2000)*.

M. Grotschel W. Dalitz and J. Lugger. 1997. Information services for mathematics and the internet. In A. Sydow, editor, *Proceedings of the 15th IMACS World Congress on Scientific Computation: Modelling and Applied Mathematics*. Wissenschaft und Technik Verlag.