In Proceedings of the International Workshop on
Multimedia Information Systems (MIS 2002),
October 2002, Tempe, Arizona

The START Multimedia Information System: Current Technology and Future Directions

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

Abstract

To address the problem of information overload in today's world, we have developed START, a natural language question answering system that provides users with high-precision multimedia information access through the use of natural language annotations. To address the difficulty of accessing large amounts of heterogeneous data, we have developed Omnibase, which assists START by integrating structured and semistructured Web databases into a single, uniformly structured "virtual database." Our ultimate goal is to develop a computer system that acts like a "smart reference librarian," and we believe we have laid a firm foundation for achieving our goal. This paper describes our current implemented system and discusses future research directions.

1 Introduction

The explosive growth of information available electronically in the form of text, images, and multimedia has given people potential access to more information than they have ever had before. However, much of this potential remains unrealized due to the lack of effective information access methods to help people separate useful knowledge from useless data.

We believe that natural language is the best information access mechanism for humans. It is intuitive, easy to use, rapidly deployable, and requires no specialized training. A step in that direction is question answering (QA), where a computer responds directly to natural language questions posed by the user. When asked "What country in Africa has the largest population," a computer should be able to respond with something like "Nigeria, with a population of 126 million, is the most populous African nation." Similarly, the computer should return digital images of Monet's water lilies in response to "Show me some famous paintings by Monet."

How can we build systems that provide natural language information access? The intuitive approach would be to take all available information, e.g., all the material in the Library of Congress, the entire World Wide Web, etc., analyze its content, and create a database containing representational structures that capture the "meaning" of the indexed material. A user question would be translated into a "semantic request," and matched against the contents of this database. Regrettably, unrestricted full-text understanding is beyond the state of the art in natural language processing, and furthermore, not all information is text; sounds, images, video, and other multimedia can all be valuable sources of knowledge. "Understanding" all these various media would require spectacular breakthroughs in other areas of artificial intelligence, such as object recognition, scene analysis, speech transcription, etc. In short, we are still years away from machines capable of distilling "meaning" from various types of multimedia documents.

Faced with the limitations of current technology and the insatiable thirst of users for more knowledge, what can we do? Rather than waiting for systems to be developed that can "understand" all available knowledge in various formats, we could instead teach the computer *where* and *how* to find the right pieces of knowledge. Such a system would act much like a librarian in the reference section of a library; although she might not be able to answer a question directly, the librarian would nevertheless be helpful because she knows where to find the relevant knowledge. In a sense, we need to give our systems *knowledge about the knowledge*.

2 Natural Language Annotations

How can we create a computer system that acts like a smart reference librarian? Our solution is natural language annotations (Katz, 1997), which are machine-parseable sentences and phrases that describe the content of various information segments. They serve as metadata describing the types of questions that a particular piece of knowledge is capable of answering. We have implemented this technology in START[1] (Katz, 1988; Katz, 1997), the first natural language question answering system available on the World Wide Web.

To illustrate how our system works, consider the HTML fragment about Olympus Mons presented in

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

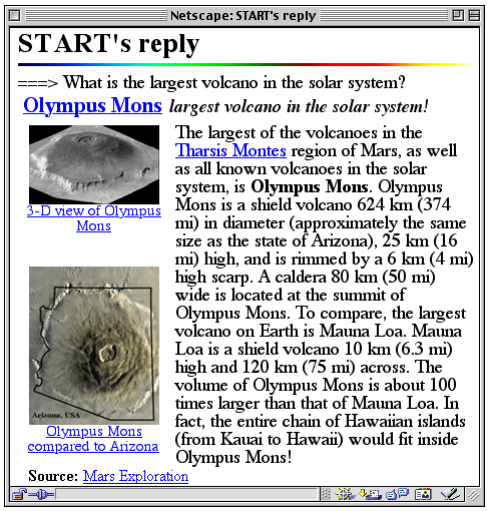

Figure 1: START responding to a question with an information segment containing both text and images.

Figure 1. It may be annotated with the following English sentences and phrases:

Mars' highest point
Largest volcano in the solar system
Olympus Mons is very tall.

START parses these annotations and stores the parsed structures (embedded ternary expressions (Katz, 1988)) with pointers back to the original information segment. To answer a question, the user query is compared against the annotations stored in the knowledge base. Because this match occurs at the level of syntactic structures, linguistically sophisticated machinery such as synonymy/hyponymy, ontologies, and structural transformation rules are all brought to bear on the matching process. Linguistic techniques allow the system to achieve capabilities beyond simple keyword matching, for example, handling complex syntactic alternations involving verb arguments. If a match is found between ternary expressions derived from annotations and those derived from the query, the segment corresponding to the annotations is returned to the user as the answer. For example, the annotations above allow START to answer the following questions (see Figure 1):

What is the highest point on Mars?
Do you know anything about Olympus Mons?
How tall is Olympus Mons?
Tell me how big Olympus Mons is.

An important feature of the annotation concept is that any information segment can be annotated: not only text, but also images, multimedia, and even procedures! For example, pictures of famous people or flags of countries in the world could be annotated with appropriate phrases and retrieved in response to user queries (Figure 2). One can also annotate a procedure for calculating distances between two locations or a procedure for calculating the current time in any world city (Figure 3).

Figure 2: A response including an image.

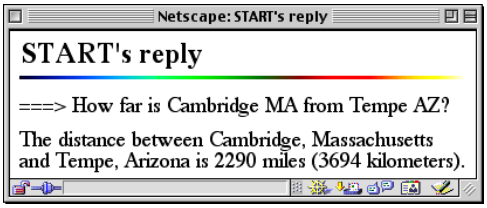

Figure 3: A response requiring calculation.

Since it came on-line in December, 1993, START has engaged in exchanges with hundreds of thousands of users all over the world, supplying them with useful knowledge.

3 Structured Knowledge

The ability to respond to natural language questions with textual and multimedia content crucially depends on natural language annotations. The knowledge coverage of the START system is thus dependent on the amount of annotated material. To increase the effectiveness of our technology, we have adapted natural language annotations to work with structured and semistructured data.

If someone is asked a question like "When did Rutherford Hayes become president of the U.S.?", he or she might locate a resource with the answer—say, a book on famous people, or a Web site about presidents—find the section for Rutherford B. Hayes, and look up the date of his inauguration. Millions of questions can be answered by following this same recipe: extract an *object* (Rutherford Hayes) and a *property* (presidential term) from the question, find a data source (e.g., the POTUS Web site) for that type of object, look up the object's Web page, and extract the *value* for the answer (see Figure 4). By generalizing such plans and integrating them into a question answering system, we can achieve information access with broad coverage.

The three main difficulties in getting a computer to answer such questions are understanding the question, identifying where to find the information, and fetching the information itself. START's parser is responsible for understanding user questions and translating them into structured queries. To help START address the other issues, we have developed a system called Omnibase (Katz et al., 2002), a "virtual" database that provides a uniform abstraction layer over multiple Web knowledge sources. Omnibase is capable of executing the structured queries

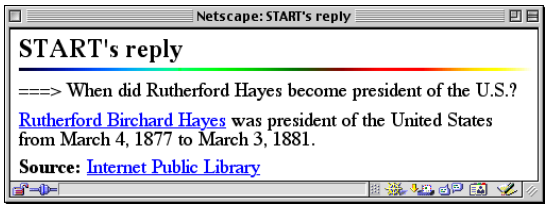

Figure 4: START answering a question with data from Omnibase, presented within a generated sentence.

generated by START. The following two sections will describe Omnibase in more detail.

4 The Web as a Database

Although the Web is predominantly comprised of unstructured static documents, pockets of structured knowledge exist, capable of providing answers to a large number of questions. For example, the CIA World Factbook provides political, geographic, and economic information about every country in the world; Biography.com contains profiles for thousands of famous (and not-so-famous) people; the Internet Movie Database contains entries for hundreds of thousands of movies, including information about their cast, production staff, etc.

Omnibase serves as a structured query interface to heterogeneous data on the World Wide Web. It is of course impossible to impose any uniform schema on the entire Web. Instead, Omnibase adopts a stylized relational model which we call the "object–property–value" (OPV) data model. Under this framework, data sources contain *objects* which have *properties*, and questions are translated into requests for the *value* of these properties.

Natural language commonly employs an 'of' relation or a possessive to express the relationship between an object and its property, e.g., "the director of La Strada" or "La Strada's director". Table 1 shows, however, that there are many alternative ways to ask for the value of an object's property.

Clearly, many other possible types of queries do not fall into the OPV model, such as questions about the relation between two objects (e.g., "How can I get from Boston to New York?").[2] However, our experiments reveal that in practice questions of the OPV type occur quite frequently. For example, just ten Web sources fashioned in the OPV manner turned out to be sufficient for handling 47% of TREC-2001 QA Track questions.

5 From Language to Knowledge

To actually answer user questions, the gap between natural language questions and structured Omnibase queries must be bridged. Natural language annotations serve as the enabling technology that allows the integration of START and Omnibase.

[2]START, however, is capable of handling such questions in a more ad-hoc fashion.

Suppose the user asks "Who directed gone with the wind?" A natural language system cannot analyze this question without first knowing that "Gone with the Wind" can be treated as a single lexical item—otherwise, the question would make no more sense than, say, "Who hopped flown down the street?" Omnibase identifies the names of objects and the data sources they are associated with; for example, "Good Will Hunting" comes from a movie data source, "United States" comes from a country data source, etc. Not only does this help START understand the user question (which can now be read as "who directed X"), but it also lets START know what data source contains the required information.

Since annotations can describe arbitrary fragments of knowledge, there is no reason why they can't be employed to describe Omnibase queries. In fact, annotations can be parameterized, i.e., they can contain symbols representative of an entire class of objects. For example, the annotation "a person wrote the screenplay for `imdb-movie`" can be attached to an Omnibase query that retrieves the writers for various movies from the Internet Movie Database (IMDb). Note that because Omnibase has knowledge of movies, it can tell START which lexical items are actually movies. The symbol `imdb-movie` serves as a placeholder for any one of the hundreds of thousands of movies that IMDb contains information about; when the annotation matches the user question, the actual movie name is instantiated and passed along in the Omnibase query.

Thus, with help from Omnibase, START translates user queries into a structured request (in the object–property–value model):

```
(get "imdb-movie"
     "Gone with the Wind (1939)"
     "DIRECTOR")
```

In this case, our natural language system needed to figure out that the user is asking about the DIRECTOR property of the object `"Gone with the Wind (1939)"`, and that this information can be found in the data source `imdb-movie`.

Omnibase looks up the data source and property to find an associated script and applies the script in order to retrieve the property value for the object. The execution of the `imdb-movie` DIRECTOR script involves looking up a unique identifier for the movie (stored locally), fetching the correct page from the IMDb Web site (via a CGI interface), and matching a textual landmark on the page (literal text and HTML tags) to find the director of the movie. As a result, the list of movie directors is returned. START then assembles the answer and presents it to the user either as a fragment of HTML or couched in natural language (Figure 5).

Currently, our system answers millions of natural language questions about places (e.g., cities, countries, lakes, coordinates, weather, maps, demograph-

Question	Object	Property	Value
Who wrote the music for Star Wars?	Star Wars	composer	John Williams
Who invented dynamite?	dynamite	inventor	Alfred Nobel
How big is Costa Rica?	Costa Rica	area	51,100 sq. km.
How many people live in Kiribati?	Kiribati	population	94,149
What languages are spoken in Guernsey?	Guernsey	languages	English, French
Show me paintings by Monet.	Monet	works	[images]

Table 1: Some sample questions that can be handled by an object–property–value model of Web data.

ics, political and economic systems), movies (e.g., titles, actors, directors), people (e.g., birth dates, biographies), dictionary definitions, and much, much more. Because START performs sophisticated syntactic and semantic processing of questions to pinpoint the exact information need of a user, questions can be answered with remarkable precision. In the period from January, 2001 to March, 2002, START and Omnibase replied to over 326 thousand queries from users all over the world. Of those, 67% were answered successfully by our system (59% of the questions answered were handled by Omnibase).

6 Beyond START and Omnibase

Despite the effectiveness of START and Omnibase in solving user information needs, there are still three major unsolved challenges:

- **The Scaling Problem.** The sheer amount of information available in the world today places a practical limit on the amount of knowledge that can be incorporated into a system by a single research group. Although natural language annotations are easy and intuitive, there is simply too much content.
- **The Knowledge Engineering Bottleneck.** Manual knowledge engineering is required to expand our system's knowledge coverage; integrating Web sources under Omnibase requires site-specific wrapper scripts. Consequently, only trained individuals can add knowledge to START and Omnibase.
- **The Fickle Web Problem.** An undesirable side-effect of the Web's dynamic nature is instability of site layout and page content. This poses a serious problem to wrapper scripts custom-tailored to specific formats. Often, dramatic changes to a page's content or layout structure will require significant modification of associated scripts.

To address these challenges, we are currently pursuing five potential solutions:

1. **Large-scale indexing via partial parsing.** Although open-domain full-text understanding is still beyond present state-of-the-art, we hope to utilize more robust shallow processing techniques to reap some of the benefits of linguistic analysis at a much larger scale.
2. **Distributed annotations.** By providing a collaborative framework in which ordinary users can annotate data, we hope to distribute and parallelize the annotations process.
3. **Leveraging Semantic Web technology.** Semantic Web research provides many possible solutions to the data integration problem.
4. **Self-repairing scripts.** We are experimenting with self-aware and self-repairing wrapper scripts that can recognize their own failures and automatically take corrective measures.
5. **Conceptual segmentation of Web pages.** Would it be possible for computers the recognize the layout of a Web page, much in the same manner as a human?

6.1 Large-scale Syntactic Indexing

Although full syntactic and semantic analysis of open-domain natural language text is beyond current technology, we believe that it is possible to augment START's manual-annotation-based approach with automatically built annotations by extracting a limited subset of relations from unstructured text; in short, information retrieval on the level of relations. This approach is promising because it attempts to address the well-known shortcomings of standard "bag-of-words" information retrieval techniques without requiring manual intervention.

To this end, we have developed Sapere, a prototype question answering system based on matching syntactic relations derived from the question with those derived from the corpus (Lin, 2001). These relations are simplified versions of START's ternary expressions (Katz, 1997), but can be generated automatically and indexed on a large scale.

We have evaluated Sapere against existing IR-based question answering systems using a restricted query set on World Book Encyclopedia. Initial results point to a dramatic improvement in precision. As an example, consider a sample output from a standard IR-based system:

> **(Q1) What do frogs eat?**
> (R1) Some bats catch fish with their claws, and a few species eat lizards, rodents, birds, and frogs.
> (R2) Bowfins eat mainly other fish, frogs, and crayfish.
> (R3) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.

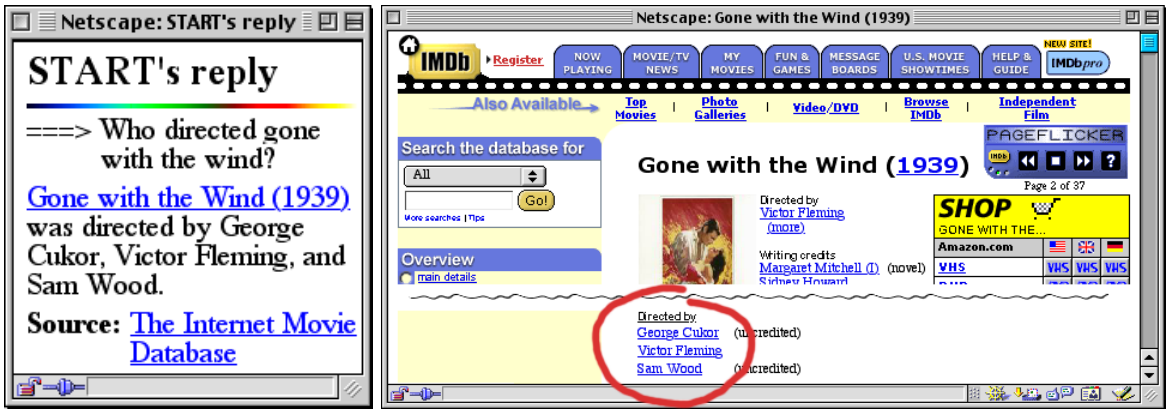

Figure 5: START's response to "Who directed Gone with the Wind" and the original Web page from which Omnibase extracts the answer.

> ...
> (R32) Kookaburras eat caterpillars, fish, frogs, insects, small mammals, snakes, worms, and even small birds.

Of the 32 sentences returned, only (R3) correctly answers the user query; the other results answer a different question—"What eats frogs?" A bag-of-words approach cannot differentiate between a query in which the frog is in the subject position and a query in which the frog is in the object position. Compare this to the results produced by Sapere:

> **(Q1) What do frogs eat?**
> (R3) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.

By examining subject-verb-object relations, Sapere can filter out irrelevant results and return only the correct responses.

6.2 Distributed Annotations

A salient feature of annotation technology is its simplicity. Because information segments can be described in everyday English, ordinary users with no technical skills qualify as annotators. As a result, the knowledge engineering bottleneck could be overcome by employing a large number of annotators working in parallel. We believe that the Web provides a low-cost mechanism for accomplishing this: by offering a collaborative framework in which users could annotate knowledge, we can distribute the knowledge engineering task across millions of people all over the world. We proposed this mechanism for gathering annotations from the Web in 1997 (Katz, 1997), and recently we have begun to implement the idea (Katz et al., 2001; Katz and Lin, 2002a).

There are two ways of setting up this collaborative framework. We could create a centralized site that people specifically visit to teach our system new nuggets of knowledge, in the spirit of Open Source projects like *Dmoz*, the Open Directory Project. As an alternative, the annotation process could be integrated into the normal browsing behavior of users, e.g., by providing an "Annotate This!" button on their browser, with which they could describe interesting information segments they encounter.

6.3 The Semantic Web

The vision of the Semantic Web (Berners-Lee et al., 2001) is to imbue Web documents with machine-understandable metadata so that software agents could more effectively utilize content. Since Semantic Web research is fundamentally attempting to address the problem of information access, we believe that many synergistic opportunities exist in the integration of our natural language technology with the Semantic Web. Question answering technology can supply an intuitive access mechanism for accessing multimedia content, and Semantic Web research has yielded many potential solutions for managing heterogeneous content.

Our proposal centers around integrating natural language annotations technology with the Resource Description Framework (RDF), the foundation of the Semantic Web. By integrating formal ontologies with natural language, we can achieve both human accessibility and computer readability. We have concretely described three separate mechanisms for marrying natural language annotations and RDF, and have built a prototype Semantic Web question answering system (Katz and Lin, 2002b).

6.4 Self-repairing Scripts

The execution of an Omnibase query involves the application of site-specific scripts that fetch the relevant Web page for an object and extract the value for a particular property. Although most of these scripts are relatively simple (e.g., simple regular expression matches), the proliferation of scripts presents a complex maintenance challenge. This problem is exacerbated by occasional changes in the layout and content of Web sources, which if too radical can necessitate the rewriting of relevant scripts.

We believe that application of machine learning

techniques can address the fragility problem associated with wrapper scripts. There have been some initial attempts to automate the wrapper generation process using machine learning, e.g., (Kushmerick et al., 1997; Muslea et al., 1999). By providing the system with known examples of properties and values, scripts can be induced automatically. However, we wish to go beyond automatic script induction and develop ways of creating self-aware and self-repairing wrappers scripts. A correctly operating script could begin to accumulate sample queries and responses, to serve as a knowledge base to recognize script failures. If the output of a script were to deviate significantly from expected output, the system would trigger a repair attempt. Because the system already has a large memory of the expected answers for various questions, this accumulated knowledge could be applied to create a new script, via machine learning techniques, for example. Only if this fails would manual intervention be necessary.

Although these ideas are still in the conceptual stage, we believe that they offer a potential buffer against dynamically changing information.

6.5 Conceptual Segmentation

We are also investigating other solutions to the knowledge engineering bottleneck. An initial and obvious solution is to facilitate the data integration process through well-designed authoring tools (e.g., (Adelberg, 1998; Sahuguet and Azavant, 1999)). By automatically preprocessing Web pages to bring potentially relevant sections to the attention of the data integrator, an authoring tool could drastically simplify the knowledge engineering process.

A more general-purpose solution to the data integration problem is to develop systems that are more "aware" of the content and semantics of the page. For example, the computer should contain heuristics regarding ways in which humans typically organize and display information, e.g., a bold heading and accompanying text frequently imply a coherent knowledge segment, a table is often used to organize properties and values, etc. In short, we want a system that can "conceptually segment" a Web page into coherent components automatically. Such a system (see Figure 6) would assist greatly in the data integration process, e.g., by presenting humans with better guesses about page layout in the context of an authoring tool, by simplifying the machine learning task, etc. We have already experimented with such tools; LaMeTH (Katz and others, 1999) is a system that attempts to recognize the conceptual structure of a Web page, not merely its HTML encoding. It provides a scripting interface that allows humans to describe content in terms of layout elements, e.g., paragraphs, lists, and tables, instead of HTML code. By developing "content-aware" systems, we hope to alleviate the problem of data integration.

7 Related Work

The use of natural language interfaces to access databases can be traced back to the sixties and seventies (Green and others, 1961; Hendrix, 1977); for a survey see (Androutsopoulos and others, 1995). Early research concentrated on adding natural language querying capabilities to existing relational databases. For the most part, the data was homogeneous and textual.

The idea of applying database techniques to the Web is not new. Many existing systems, e.g., ARANEUS (Atzeni and others, 1997), ARIADNE (Knoblock and others, 1999), Information Manifold (Kirk and others, 1995), LORE (McHugh and others, 1997), TSIMMIS (Hammer and others, 1997), have attempted to unify heterogeneous Web sources under a common interface. Unfortunately, queries to such systems must be formulated in SQL, Datalog, or some similarly formal language, which render them inaccessible to the average user. Because the focus of research in semistructured data has been on issues such as the modeling of heterogeneous knowledge sources, the expressiveness of the query language, and implementation issues arising from the unreliable nature of the Web, little work has been done on natural language querying capabilities.

What makes START and Omnibase unique among these systems is natural language question answering abilities and its use of the object–property–value data model. By allowing ordinary users to ask questions in English, we provide intuitive and precise information access to a wealth of information. Furthermore, since our data model corresponds naturally to both user questions and on-line content, the data integration task becomes more intuitive.

8 Conclusion

START and Omnibase are complementary components of a question answering system that addresses users' information access needs. START understands natural language questions and retrieves multimedia answers via annotations. Omnibase helps START translate natural language questions to structured queries. It serves as an abstraction layer which lets START treat heterogeneous Web sources as a uniform "virtual database". By providing a uniform natural language interface to heterogeneous knowledge on the World Wide Web, we can supply users with "just the right information." We believe that structured access to heterogeneous on-line data sources should be a key component of any future natural language question answering system.

Our systems have proven to be a success by a variety of measures. Nevertheless, many directions for future research remain unexplored, some of which we have discussed in this paper. We remain optimistic about the future prospects of natural language information access systems.

Figure 6: A system that "conceptually segments" knowledge fragments on a Web page. Here we see five knowledge fragments, numbered and demarcated by boxes, which the system has detected.

9 Acknowledgements

This research is funded by DARPA under contract number F30602-00-1-0545 and administered by the Air Force Research Laboratory. Additional funding is provided by the Oxygen Project.

References

B. Adelberg. 1998. NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. *SIGMOD Record*, 27:283–294.

I. Androutsopoulos et al. 1995. Natural language interfaces to databases—an introduction. *Natural Language Engineering*, 1(1):29–81.

P. Atzeni et al. 1997. Semistructured and structured data in the Web: Going back and forth. In *Workshop on Management of Semistructured Data at PODS/SIGMOD'97*.

T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American*, 284(5):34–43.

B. Green et al. 1961. BASEBALL: An automatic question answerer. In *Proceedings of the Western Joint Computer Conference*.

J. Hammer et al. 1997. Extracting semistructured information from the Web. In *Workshop on Management of Semistructured Data at PODS/SIGMOD'97*.

G. Hendrix. 1977. Human engineering for applied natural language processing. Technical Note 139, SRI International.

B. Katz and J. Lin. 2002a. Annotating the Semantic Web using natural language. In *Proceedings of the 2nd Workshop on NLP and XML*.

B. Katz and J. Lin. 2002b. Natural language annotations for the Semantic Web. In *Proceedings of ODBASE 2002*.

B. Katz et al. 1999. Integrating large lexicons and Web resources into a natural language query systen. In *Proceedings of IEEE ICMCS '99*.

B. Katz, J. Lin, and S. Felshin. 2001. Gathering knowledge for a question answering system from heterogeneous information sources. In *Proceedings of the ACL 2000 HLT Workshop*.

B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G. Marton, A.J. McFarland, and B. Temelkuran. 2002. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of NLDB 2002*.

B. Katz. 1988. Using English for indexing and retrieving. In *RIAO '88*.

B. Katz. 1997. Annotating the World Wide Web using natural language. In *RIAO '97*.

T. Kirk et al. 1995. The Information Manifold. Technical report, AT&T Bell Laboratories.

C. Knoblock et al. 1999. The Ariadne approach to Web-based information integration. *International Journal on Cooperative Information Systems*, 10(1/2):145–169.

N. Kushmerick, D. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. In *IJCAI-97*.

J. Lin. 2001. Indexing and retrieving natural language using ternary expressions. Master's thesis, Massachusetts Institute of Technology.

J. McHugh et al. 1997. Lore: A database management system for semistructured data. Technical report, Stanford University Database Group.

I. Muslea, S. Minton, and C. Knoblock. 1999. A hierarchical approach to wrapper induction. In *3rd International Conference on Autonomous Agents*.

A. Sahuguet and F. Azavant. 1999. WysiWyg Web Wrapper Factory. In *Proceedings of WWW8*.