Chapter 17

Viewing the Web as a Virtual Database for Question Answering

Boris Katz, Sue Felshin, Jimmy Lin, and Gregory Marton
MIT Computer Science and Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139
{boris,sfelshin,jimmylin,gremio}@csail.mit.edu

Abstract

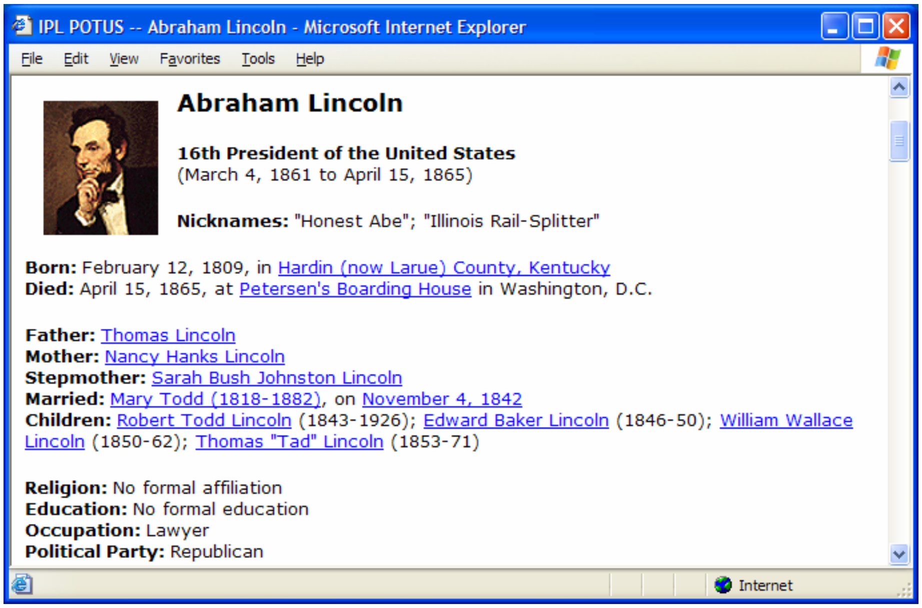
Although the World Wide Web contains a tremendous amount of information, the lack of intuitive information access methods and the paucity of uniform structure make finding the right knowledge difficult. Our solution is to turn the Web into a "virtual database" and to access it through natural language. We have accomplished this by developing a stylized relational framework, called the *object-property-value* model, which captures the regularity found in both natural language questions and Web resources. We have adopted this framework in START and Omnibase, two components of a system that understands natural language questions and responds with answers extracted on the fly from heterogeneous and semistructured Web sources. Our system can answer millions of questions from hundreds of Web resources with high precision.

1. Introduction

The vast amount of information available on the World Wide Web has given people potential access to more knowledge than they have ever had before. But, much of this potential remains unrealized due to the lack of effective information access methods to help people separate useful knowledge from useless data.

Question answering has recently emerged as a technology that promises to provide more intuitive methods of information access. In contrast to the traditional information retrieval model of formulating queries and browsing resulting documents, a question answering system accepts user information requests phrased in everyday language and responds with a concise answer. When asked "What country in Asia has the lowest infant mortality rate?", a

The final version of this book chapter appears as:
Boris Katz, Sue Felshin, Jimmy Lin, and Gregory Marton. Viewing the Web as a Virtual Database for Question Answering. In Mark T. Maybury, editor, New Directions in Question Answering. Cambridge, Massachusetts: MIT Press, 2004, pages 215-226.


IPL POTUS -- Abraham Lincoln - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Internet

Abraham Lincoln

16th President of the United States
(March 4, 1861 to April 15, 1865)

Nicknames: "Honest Abe"; "Illinois Rail-Splitter"

Born: February 12, 1809, in Hardin (now Larue) County, Kentucky
Died: April 15, 1865, at Petersen's Boarding House in Washington, D.C.

Father: Thomas Lincoln
Mother: Nancy Hanks Lincoln
Stepmother: Sarah Bush Johnston Lincoln
Married: Mary Todd (1818-1882), on November 4, 1842
Children: Robert Todd Lincoln (1843-1926); Edward Baker Lincoln (1846-50); William Wallace Lincoln (1850-62); Thomas "Tad" Lincoln (1853-71)

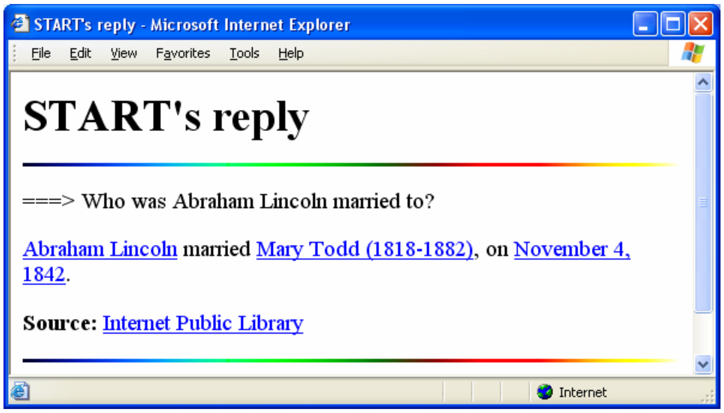
Religion: No formal affiliation
Education: No formal education
Occupation: Lawyer
Political Party: Republican

Figure 1: A page from the Internet Public Library about an *object* (Abraham Lincoln), which contains *values* for many *properties*.

computer system should be able to respond with something like "Singapore has the lowest infant mortality rate among countries in Asia (3.62 deaths per 1000 births)." Similarly, the computer should be able to return digital images of Monet's water lilies in response to "Show me some famous paintings by Monet."

In this chapter, we present a data model, called the *object-property-value model*, for organizing and integrating heterogeneous, semistructured, and structured resources. Because our data model naturally captures the semantic content of many real-world user questions, it serves as a powerful paradigm for building question answering systems. We have implemented our ideas in START and Omnibase, two components of a Web-based question answering system.

Our data model can be illustrated through a simple scenario: suppose someone is asked a question like "Who was Abraham Lincoln married to?". He or she might locate a resource with the answer—say, a book on famous historical figures, or a


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Internet

START's reply

===> Who was Abraham Lincoln married to?

Abraham Lincoln married Mary Todd (1818-1882), on November 4, 1842.

Source: Internet Public Library

Figure 2: START answering a question using Omnibase.

Web site about presidents—find the section for Abraham Lincoln (Figure 1), and look up information about his spouse. Millions of questions can be answered in this manner: by extracting an *object* (Abraham Lincoln) and a *property* (spouse) from the question, finding a data source (e.g., the Internet Public Library) for that type of object, looking up the object's Web page, and extracting the *value* for the answer (Mary Todd). Our question answering system responds to natural language questions using exactly this procedure (Figure 2).

The three main challenges in getting a computer to answer natural language questions are understanding the question, identifying where to find the information, and fetching the information itself. START (Katz, 1988; Katz, 1997) and Omnibase (Katz et al., 2002a) comprise our natural language question answering system[1] developed to address these challenges. START is responsible for understanding user questions and translating them into structured object-property-value queries. Omnibase is a "virtual" database that provides a uniform interface to multiple Web knowledge sources, and is capable of executing the structured queries generated by START.

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

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
When did Sweden gain its independence?	Sweden	independence	June 6, 1523
Show me paintings by Monet.	Monet	works	[images]

Table 1: Sample questions captured by our object-property-value model.

2. The Web as a Database

The World Wide Web contains numerous resources that hold vast amounts of knowledge in relatively structured formats. For example, the World Factbook provides political, geographic, and economic information about every country in the world; Biography.com has collected profiles of over twenty-five thousand famous people; the Internet Movie Database stores entries for hundreds of thousands of movies, including information about their cast, directors, and other properties. Many of these sources are part of the “deep” or “invisible” Web, which cannot be accessed through normal hypertext navigation; such knowledge is stored in databases accessible only through specific search interfaces.

To effectively use these semistructured resources for question answering, the plethora of knowledge sources must be integrated under a common interface or query language. Our Omnibase system accomplishes just this by acting like a “virtual database”: the system presents a local and uniform view of content distributed across remote servers in heterogeneous formats. Diverse resources are captured using our object-property-value data model. Under this framework, data sources contain *objects* which have *properties*, and questions are translated into requests for the *value* of these properties. Omnibase serves as a mediator between a structured query interface and disparate resources. It is of course impossible to impose any uniform schema on the entire Web. However, we believe that our object-property-value model is simple enough to capture the content of many Web resources, and expressive enough to answer many types of natural language questions.

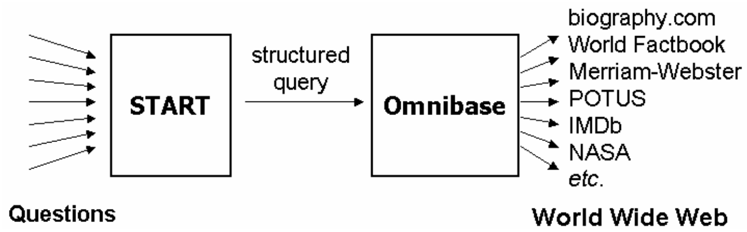


Figure 3: Overall architecture of START and Omnibase.

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. Often, properties can be encoded in the arguments of verbs, e.g., the subject of the verb *invent* is the value of the *inventor* property, and the direct object of the verb is the value of the *invention* property. Adjectives can similarly be interpreted as properties, e.g., "how big" is understood as requesting the *area* property of a particular country.

Clearly, many other possible types of queries do not fall into the object-property-value 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 object-property-value type occur quite frequently. For example, just ten Web sources organized according to our data model suffice to answer 27% of TREC-9 and 47% of TREC-10 questions from the QA Track (Lin, 2002).

3. START and Omnibase

A schematic representation of the relationship between START and Omnibase is shown in Figure 3. Conceptually, START is responsible for understanding the syntax and the semantics of user questions and then distilling the information

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

requests into object-property-value queries. Omnibase is responsible for executing those queries and fetching answers from the World Wide Web and other sources. Omnibase serves as an abstraction layer over collections of heterogeneous semistructured data, providing START with a uniform query interface. The following sections describe in more detail how START and Omnibase interact to form two components of our question answering system.

3.1 Analyzing Natural Language

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 constituents of a sentence, but the three most salient of them, the subject of a sentence, the object, and the relation between them are singled out as playing a special role in indexing. These constituents are explicitly represented in a discrimination network for efficient retrieval. As a result, all sentences analyzed by START are indexed as ternary expressions (T-expressions), **<subject relation object>** (Katz, 1988). Certain other constituents (adjectives, possessive nouns, prepositional phrases, etc., are used to create additional T-expressions in which prepositions and several special words may serve as relations. Additional grammatical features associated with a sentence—adverbs and their position, tense, auxiliaries, voice, negation, etc.—are recorded in a separate representational structure called a *history*. When we index the T-expression in the knowledge base, we cross-reference its 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. Thus, the system 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.

The embedded ternary expressions used by START 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 surface syntax but close in meaning are not considered similar by the system. For example, speakers of English would recognize that "The president surprised the country with his determination." and "The president's determination surprised the country." have the same meaning, despite differences in their syntactic structures. This poses a problem to any syntactically-based representation of language: due to different syntactic structures, a machine cannot automatically match a question stated one way with an answer formulated in another way.

To be able to handle such phenomena, commonly known as lexical alternations, a natural language system should be aware of the interactions between the syntactic and semantic properties of verbs. The *surprise* example above is just one example of the alternations phenomena that pervade natural language. In this instance, we want START to know that whenever *A surprised B with C*, then it is also true that *A's C surprised B*. We do this by introducing structural transformational rules, called S-rules (Katz and Levin, 1988), that make explicit the relationship between alternate realizations of verb arguments. Linguists have noticed that verbs which undergo the same alternations can be grouped into semantic classes (Levin, 1993); as a result, S-rules can be generalized to cover entire classes of verbs. Thus, a relatively small number of such rules suffice to capture a significant portion of the phenomena. In the example above, the verb *surprise* is a member of the *emotional reaction* class, whose members also include other verbs like *amaze*, *amuse*, *impress*, *scare*, *stun*, etc. Although S-rules must be manually constructed, a special component inside START allows the rules to be inferred from a set of English sentences which capture a specific instance of the rule. START analyzes these sentences, queries the user for additional information regarding elements of corresponding T-expressions, and then builds appropriately-generalized S-rules automatically.

3.2 Natural Language Annotations

Mediation between natural language questions and Omnibase queries is accomplished through a technology we developed called *natural language annotations* (Katz, 1997), which are machine-parseable sentences and phrases that describe the content of various content segments. They serve as metadata describing the questions that a particular piece of knowledge can answer.

In the simplest mode of operation, annotations are attached to static content segments (e.g., a part of an HTML document, an image, etc.). The combination of natural language annotations and a content segment forms a simple schema:

> **Annotations:**
> Mars' two moons
> Phobos and Deimos orbit Mars.
>
> **Content Segment:**
> [images] Mars has two small moons: Phobos and Deimos. Phobos (fear) and Deimos (panic) were named after the horses that pulled the chariot of the Greek war god Ares, the counterpart to the Roman war god Mars…

START parses the annotations and stores the parsed ternary expressions with pointers back to the original content 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 structure, 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 (Katz and Levin, 1988), recognizing active/passive variations, or dealing with such linguistic phenomena as dative movement. If a match is found between ternary expressions derived from annotations and those derived from the query, the corresponding segment is returned to the user as the answer. For example, the annotations above allow START to answer questions such as the following:

What satellites orbit Mars?
How many moons orbit Mars?
What are the names of Mars' moons?

An important feature of the annotation concept is that any information segment can be annotated: not only static content, but also procedures or database queries. For example, one can annotate a procedure for calculating distances between two locations. In this case, the matching of a user question with an annotation triggers the extraction of relevant parameters, i.e., the name of the two locations. These extracted elements become dynamically instantiated parameters in the procedure attached to the natural language annotation. The execution of that procedure results in the computation of the answer. Similarly, START is connected to Omnibase through schemata annotated with natural language annotations.

3.3 Identifying Objects and Sources

Suppose a user asks "Who directed gone with the wind?" START 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 serves as a large external lexicon that helps START identify the names of objects and the data sources they are associated with.

Knowledge about objects not only helps START understand the user question (which can now be read as "Who directed *x*?"), but also lets START know what data source contains the information; i.e., information about *x* is contained in a movie database. With this information, START can construct an Omnibase query that, when executed, will compute the answer to the user question.

3.4 Fetching Answers

Omnibase contains a large collection of wrapper scripts tailored to specific Web resources and other data sources. With this framework, our system is able to abstract away the idiosyncrasies of each individual source and

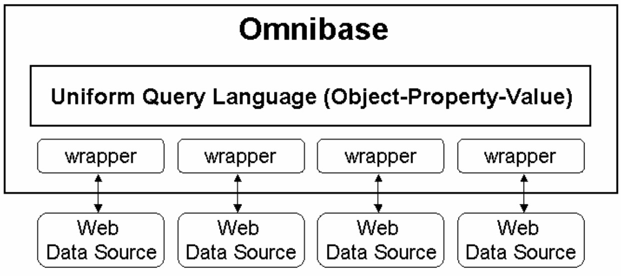


Figure 4: Organization of Omnibase.

present START with a uniform query interface under the object-property-value model (Figure 4).

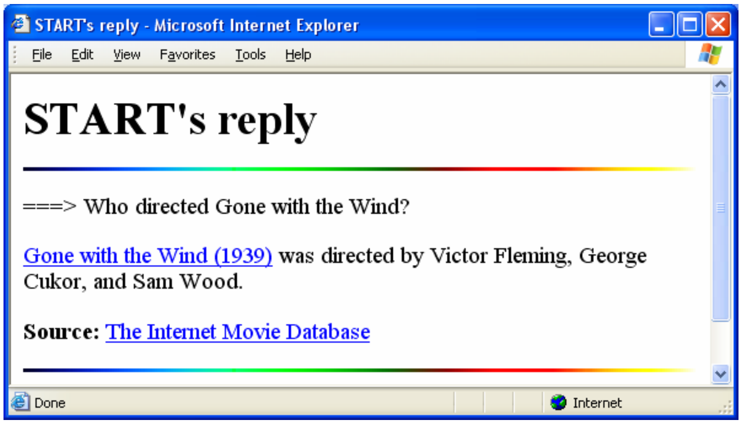
For some sources, pages about individual objects exist as static HTML pages, but for other sources, pages are generated dynamically with a CGI script. Depending on the source, values of properties might be aligned in tables, in lists, or even in free text. Typically, values are extracted by performing regular expression pattern matching on the source HTML page using Omnibase's built-in facilities for pattern matching.

Once again, consider the question “Who directed gone with the wind?”. After “Gone with the Wind” is identified as the name of a movie from the Internet Movie Database, the question then matches a particular natural language annotation, resulting in a complete object-property-value query:

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

To execute the query, Omnibase looks up the data source and property to find the associated wrapper script and applies the script to the object 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 Website, and matching a textual landmark on the page to find the director of the movie. As a result, the list of movie directors is returned:

```
("George Cukor" "Victor Fleming" "Sam Wood")
```


START's reply - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
Internet

START's reply

===> Who directed Gone with the Wind?

Gone with the Wind (1939) was directed by Victor Fleming, George Cukor, and Sam Wood.

Source: The Internet Movie Database

Figure 5: START's answer to "Who directed Gone with the Wind?"

START then assembles the answer and presents it to the user either as a fragment of HTML or couched in natural language (see Figure 5). By providing the right granularity of information to the user through natural language access, we can most effectively fulfill a user's information needs.

4. Deployment and Evaluation

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. Currently, over one hundred Web resources, many of them containing hundreds of thousands of individual pages, have been integrated into START and Omnibase.

At present, our system answers millions of natural language questions about places (e.g., cities, countries, lakes, coordinates, weather, maps, demographics, 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 linguistic processing of questions to pinpoint the exact information need of a user, questions can be answered with remarkable precision.

	2000	**2001**	**2002**
Answered using Omnibase	**85k (27.1%)**	**100k (37.6%)**	**129k (37.9%)**
Answered with native KB	**123k (39.3%)**	**74k (27.9%)**	**107k (31.5%)**
Don't know	72k (22.9%)	65k (24.3%)	78k (22.8%)
Don't understand	19k (6.0%)	15k (5.5%)	14k (4.2%)
Unknown word	15k (4.8%)	12k (4.7%)	12k (3.6%)
Total	313k (100%)	266k (100%)	342k (100%)

	2000	**2001**	**2002**
Total Answered Successfully	**208k (66.4%)**	**174k (65.5%)**	**237k (69.4%)**
Answered using Omnibase	40.9%	57.4%	54.6%
Answered with native KB	59.1%	42.6%	45.4%

Table 2: Evaluation of START and Omnibase. The top table shows results of all types of system responses and the bottom table shows the performance contribution of Omnibase and START's native KB.

In the period from January 2000, to December 2002, about a million questions were posed to START and Omnibase. Of those, approximately 67% were answered successfully by our system (Table 2). The most common failure was the lack of knowledge: in those cases, START was successful in analyzing the user question, but the knowledge required to answer the question was simply not available to the system. Only in approximately 10% of the questions did START encounter unknown lexical items or fail to parse the sentence. Of all the questions successfully answered by our system, about 50% were handled by Omnibase.

We have also deployed a simplified version of our annotation-based approach to question answering in a system called Aranea, which was formally evaluated at the TREC-2002 question answering track. Our data model and annotation-based technique was able to achieve 71% accuracy in answering previously unseen questions (Lin and Katz, 2003). Furthermore, similar techniques for structuring knowledge sources to answer frequently-occurring questions have been effectively used by other TREC systems (Chu-Carroll et al., 2002; Clarke et al., 2002).

In addition to the main START Website, we have built domain-specific servers for a variety of custom applications. In September, 1989, the Voyager 2 space probe concluded its Grand Tour of the Solar System with a flyby of Neptune. In cooperation with researchers from the Jet Propulsion Laboratory (JPL), START was taught knowledge about Voyager and Neptune, and it successfully answered questions posed by members of the press during the encounter. In another collaboration with NASA, our system was on display at JPL's annual open house in May, 2001. Guests of all ages were both informed and entertained by the system. Other custom applications of START include a permanent exhibit at the MIT Museum and a prototype in which the system was taught knowledge from a college-level biology textbook.

Finally, the viability of our annotation-based question-answering technique has also been demonstrated commercially. For example, Ask Jeeves, currently the Web's second most popular search engine, has licensed certain technology pioneered by START (Katz and Winston, 1994, 1995)

5. Related Work

The use of natural language interfaces to access databases can be traced back to the 60s's and 70's in systems such as BASEBALL (Green et al., 1961), Lunar (Woods et al., 1972), and Lifer (Hendrix, 1977); see (Androutsopoulos et al., 1995) for a survey. In these systems, syntactic analysis of user questions was intertwined with the semantic interpretation process. The result was brittle, monolithic systems that were difficult to adapt to new domains. Furthermore, these early systems were applied to well-structured data originating from a single source, which severely limited the scope of possible applications. In contrast, START and Omnibase maintain strict separation of syntax and semantics through the object-property-value data model: START is capable of performing complex linguistic analysis, but distills results into a simple yet expressive semantic structure, i.e., a structured query to Omnibase.

The idea of applying database techniques to the World Wide Web is not new, either; see (Florescu et al., 1998) for a survey. Many existing systems, e.g., Araneus (Atzeni et al., 1997), Ariadne (Knoblock et al., 2001), Information Manifold (Kirk et al., 1995), Lore (McHugh et al., 1997), Tsimmis (Hammer et al., 1997), and others, have attempted to integrate heterogeneous Web sources under a common interface. Unfortunately, queries to such systems must be formulated in SQL, Datalog, or some similar formal language, which render them inaccessible to the average user. What makes START and Omnibase unique among database systems that integrate semistructured data is its use of the object-property-value data model and the natural language interface it facilitates. Because this model corresponds naturally to both user questions and online content, the data integration task becomes more intuitive. To our knowledge, the START-Omnibase combination is the first system to provide natural language access to heterogeneous and semistructured data.

6. Contributions

We have organized diverse heterogeneous and semistructured data on the World Wide Web by creating an abstraction layer centered around our object-property-value model. Because many natural language queries translate into this data model, we can capture their semantics in a simple yet expressive manner. Furthermore, since our data model is reflective of real-world user queries, broad knowledge coverage can be achieved with a reasonable amount of manual labor.

We have implemented our ideas in two components of a question answering system. START understands users' natural language queries and translates them into object-property-value queries, which are subsequently executed by Omnibase. Our experience shows this to be a very intuitive division of labor that allows easy integration of both advanced natural language processing and database techniques. Together, START and Omnibase provide access to a wealth of information freely available on the World Wide Web. We believe that our techniques for structuring heterogeneous and semistructured data sources offer an effective strategy for tackling the information access challenge and will become an integral part of future question answering systems.

7. Acknowledgments

This research was funded by DARPA under contract number F30602-00-1-0545. Additional funding was provided by the Oxygen Project.

8. References

Androutsopoulos, I.; Ritchie, G. D.; and Thanisch, P. 1995. Natural Language Interfaces to Databases—An Introduction. *Natural Language Engineering*, 1(1):29–81.

Atzeni, P.; Mecca, G.; and Merialdo, P. 1997. Semistructured and Structured Data in the Web: Going back and forth. In *Proceedings of the Workshop on Management of Semistructured Data at PODS/SIGMOD'97*.

Chu-Carroll, J.; Prager, J.; Welty, C.; Czuba, K.; and Ferrucci, D. 2002. A Multi-Strategy and Multi-Source Approach to Question Answering. In *Proceedings of TREC 2002*.

Clarke, C.; Cormack, G.; Kemkes, G.; Laszlo, M.; Lynam, T.; Terra, E.; and Tilker, P. 2002. Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002). In *Proceedings of TREC 2002*.

Florescu, D.; Levy A.; and Mendelzon, A. 1998. Database Techniques for the World-Wide Web: A Survey. *SIGMOD Record*, 27(3):59–74.

Green, B.; Wolf, A.; Chomsky, C.; and Laughery, K. 1961. BASEBALL: An Automatic Question Answerer. In *Proceedings of the Western Joint Computer Conference*.

Hammer, J.; Garcia-Molina, H.; Cho, J.; Aranha R.; and Crespo, A. 1997. Extracting Semistructured Information from the Web. In *Proceedings of the Workshop on Management of Semistructured Data at PODS/SIGMOD'97*.

Hendrix, G. G. 1977. Human Engineering for Applied Natural Language Processing. Technical Note 139, SRI International.

Katz, B. 1997. Annotating the World Wide Web using Natural Language. In *Proceedings of RIAO '97*.

Katz, B. 1988. Using English for Indexing and Retrieving. In *Proceedings of RIAO '88*.

Katz, B.; Felshin, S.; Yuret, D.; Ibrahim, A.; Lin, J.; Marton, G.; McFarland, A. J.; and Temelkuran, B. 2002a. Omnibase: Uniform Access to Heterogeneous Data for Question Answering. In *Proceedings of NLDB 2002*.

Katz, B.; and Winston, P. 1995. Method and Apparatus for Utilizing Annotations to Facilitate Computer Retrieval of Database Material, United States Patent No. 5,404,295.

Katz, B.; and Winston, P. 1994. Method and Apparatus for Generating and Utilizing Annotations to Facilitate Computer Text Retrieval, United States Patent No. 5,309,359.

Katz, B.; and Levin, B. 1988. Exploiting Lexical Regularities in Designing Natural Language Systems. In *Proceedings of COLING-1988*.

Kirk, T.; Levy A.; Sagiv, Y.; and Srivastava, D. 1995. The Information Manifold. Technical report, AT&T Bell Laboratories.

Knoblock, C.; Minton, S.; Ambite, J. L.; Ashish, N.; Muslea, I.; Philpot, A.; and Tejada, S. 2001. The Ariadne Approach to Web-based Information Integration. *International Journal on Cooperative Information Systems (IJCIS) Special Issue on Intelligent Information Agents: Theory and Applications*, 10(1/2):145–169.

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

Lin, J. 2002. The Web as a Resource for Question Answering: Perspectives and Challenges. In *Proceedings of LREC-2002*.

Lin, J.; and Katz, B. 2003. Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques. In *Proceedings of CIKM-2003*.

McHugh, J.; Abiteboul, S.; Goldman, R.; Quass, D.; and Widom, J. 1997. Lore: A Database Management System for Semistructured Data. Technical report, Stanford University Database Group.

Woods, W. A.; Kaplan, R. M.; and Nash-Webber, B. L. 1972. The Lunar Sciences Natural Language Information System: Final report. Technical Report 2378, BBN.