In Proceedings of the 7th International Workshop on
Applications of Natural Language to Information Systems
(NLDB 2002), June 2002, Stockholm, Sweden

Omnibase: Uniform Access to Heterogeneous Data for Question Answering

Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran

Artificial Intelligence Laboratory
200 Technology Square, Cambridge, MA 02139
boris@ai.mit.edu

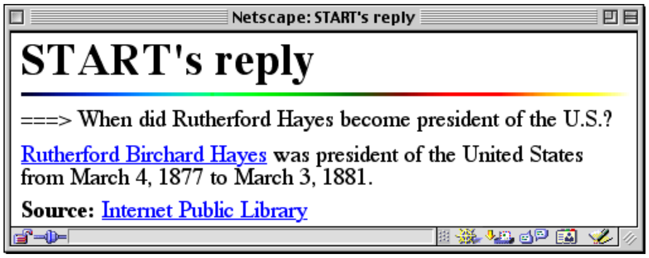
Abstract. Although the World Wide Web contains a tremendous amount of information, the lack of uniform structure makes finding the right knowledge difficult. A solution is to turn the Web into a "virtual database" and to access it through natural language. We built Omnibase, a system that integrates heterogeneous data sources using an *object–property–value* model. With the help of Omnibase, our START natural language system can now access numerous heterogeneous data sources on the Web in a uniform manner, and answers millions of user questions with high precision.

1 Introduction

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 in this manner: by extracting an *object* (Rutherford Hayes) and a *property* (presidential term) from the question, finding a data source (e.g., the POTUS Web site, `http://www.ipl.org/ref/POTUS`) for that type of object, looking up the object's Web page, and extracting the *value* for the answer (see Figure 1).

The three main challenges in getting a computer to answer such questions are understanding the question, identifying where to find the information, and fetching the information itself. START [9, 10] and Omnibase comprise our natural language question answering system[1] developed to addresses these challenges. START is responsible for understanding user questions and translating them into structured queries. Omnibase is a "virtual" database that provides a uniform interface to multiple Web knowledge sources, capable of executing the structured queries generated by START. Currently, our system can answer millions of questions about variety of topics such as almanac information (cities, countries, lakes, etc.; weather, demographics, economics, etc.), facts about people (birth dates, biographies, etc.), and so forth.

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



START's reply

===> When did Rutherford Hayes become president of the U.S.?

Rutherford Birchard Hayes was president of the United States from March 4, 1877 to March 3, 1881.

Source: Internet Public Library

Fig. 1. The START system answering a question using Omnibase.

2 Data Model

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. To address this challenge, Omnibase adopts a stylized relational model which we call the "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.

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". The following table shows, however, that there are many alternative ways to ask for the value of a property of an object.

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]

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?"). However, our experiments reveal that in practice questions of the object–property–value type occur quite frequently. For example, just ten Web sources fashioned in the object–property–value manner turned out to be sufficient for handling 37% of TREC-9 and 47% of TREC-2001 questions from the QA track [14].

3 How the System Works

Suppose the 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 identifies the names of objects and the data sources they are associated with; for example, "Good Will Hunting" comes from a movie data source, "Taiwan" 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 information, i.e., look in a movie database. The actual process involves matching syntactic structures derived from the question with those derived from natural language annotations [10]. These annotations are machine-parseable sentences and phrases that serve as metadata to describe knowledge segments (in this case, Omnibase queries). A successful match triggers the execution of an Omnibase query. Because START performs this match at the syntactic level, linguistic machinery is utilized to achieve capabilities beyond simple keyword matching, for example, complex syntactic alternation involving verb arguments. More detailed descriptions of this technology can be found in [9, 10].

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`, corresponding to the Internet Movie Database.

Omnibase looks up the data source and property to find the associated script and applies the script to the object in order to retrieve the property value for the object.[2] 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 (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:

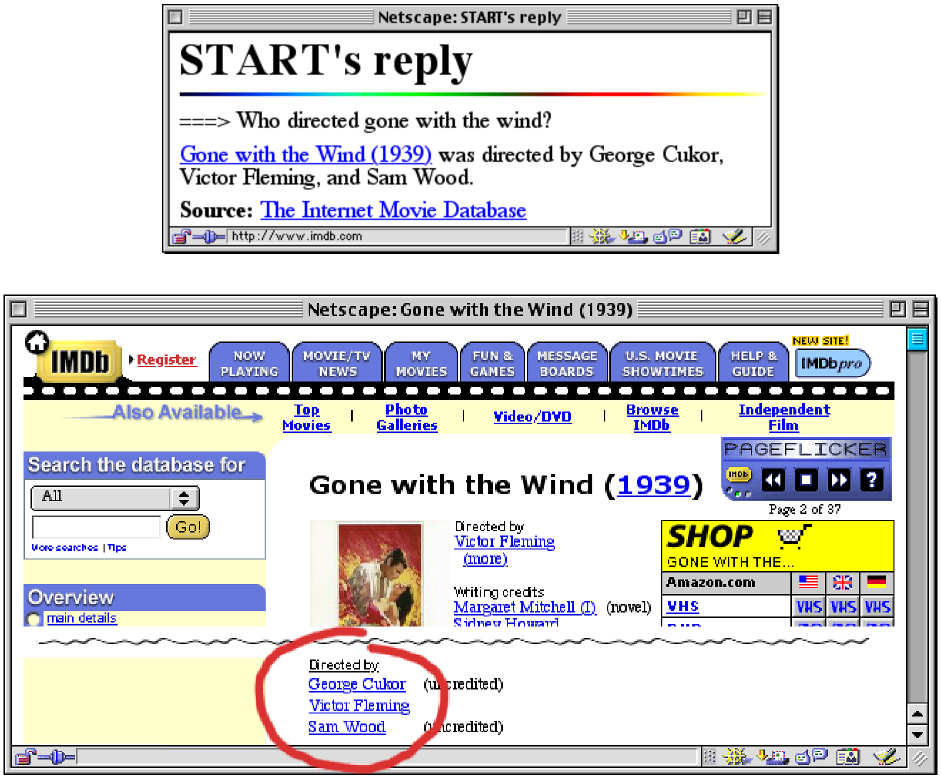
```
(get "imdb-movie" "Gone with the Wind (1939)" "DIRECTOR") =>
("George Cukor" "Victor Fleming" "Sam Wood")
```

START then assembles the answer and presents it to the user either as a fragment of HTML or couched in natural language (Figure 2).

4 Related Work

The use of natural language interfaces to access relational databases can be traced back to the sixties; for a survey see [2].

[2] Such scripts are sometimes called wrappers [6].



Netscape: START's reply

START's reply

===> Who directed gone with the wind?

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

Source: The Internet Movie Database

http://www.imdb.com

Netscape: Gone with the Wind (1939)

Gone with the Wind (1939)

Directed by
Victor Fleming
(more)

Writing credits
Margaret Mitchell (I) (novel)
Sidney Howard

Directed by
George Cukor (uncredited)
Victor Fleming
Sam Wood (uncredited)

Fig. 2. START's response to "Who directed Gone with the Wind" is shown above. The original Web page from which Omnibase extracts the answer is shown below.

The idea of applying database techniques to the World Wide Web is not new. Many existing systems, e.g., ARANEUS [3], ARIADNE [12], Information Manifold [11], LORE [15], TSIMMIS [7], 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.

A well known disadvantage of data integration systems is the manual labor involved in writing wrappers [6]. Often, wrapper generation can be expedited by a well-designed authoring tool, e.g., [1, 17]. Alternatively, machine learning techniques can automate the wrapper generation process [13, 8, 16, 5] using annotated examples. Most promising is Semantic Web [4] research; if someday it can imbue ordinary Web documents with semantic annotations, integration of multiple knowledge sources could be accomplished effortlessly.

What makes Omnibase unique among these systems is its use of the object–property–value data model; because this model corresponds naturally to both user questions and online content, the data integration task becomes more intuitive.

5 Contributions

We have built Omnibase as an abstraction layer over diverse, semi-structured, online content, centered around "object–property–value" queries. Because our data model is reflective of real-world user queries, we can achieve broad knowledge coverage with a reasonable amount of manual labor.

Omnibase has given START, our natural language question answering system, access to a wealth of information freely available on the World Wide Web. In the future, we intend to extend the data model and to automate the data integration process. We believe that structured access to online data sources will be a key component of any future natural language question answering system.

References

1. B. Adelberg. NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. *SIGMOD Record*, 27:283–294, 1998.
2. Ion Androutsopoulos, G. Ritchie, and P. Thanisch. Natural language interfaces to databases—an introduction. *Natural Language Engineering*, 1(1):29–81, 1995.
3. P. Atzeni, G. Mecca, and P. Merialdo. Semistructured and structured data in the Web: Going back and forth. In *Workshop on Management of Semistructured Data at PODS/SIGMOD'97*, 1997.
4. T. Berners-Lee. *Weaving the Web.* Harper, New York, 1999.
5. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Automatically deriving structured knowledge bases from on-line dictionaries. Technical Report CMU-CS-98-122, Carnegie Mellon University, 1998.
6. D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.
7. J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the Web. In *Workshop on Management of Semistructured Data at PODS/SIGMOD'97*, 1997.
8. C. Hsu and C. Chang. Finite-state transducers for semi-structured text mining. In *IJCAI-99 Workshop on Text Mining*, 1999.
9. B. Katz. Using English for indexing and retrieving. In *RIAO '88*, 1988.
10. B. Katz. Annotating the World Wide Web using natural language. In *RIAO '97*, 1997.
11. T. Kirk, A. Levy, Y. Sagiv, and D. Srivastava. The Information Manifold. Technical report, AT&T Bell Laboratories, 1995.
12. C. Knoblock, S. Minton, J. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada. The Ariadne approach to Web-based information integration. *International Journal on Cooperative Information Systems*, 10(1/2):145–169, 1999.
13. N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *IJCAI-97*, 1997.
14. J. Lin. The Web as a resource for question answering: Perspectives and challenges. In *LREC2002*, 2002.
15. J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A database management system for semistructured data. Technical report, Stanford University Database Group, February 1997.
16. I. Muslea, S. Minton, and C. Knoblock. A hierarchical approach to wrapper induction. In *3rd International Conference on Autonomous Agents*, 1999.
17. A. Sahuguet and F. Azavant. WysiWyg Web Wrapper Factory. In *WWW8*, 1999.