Providing Database-like Access to the Web Using Queries Based on Textual Similarity

William W. Cohen
AT&T Labs—Research
180 Park Avenue, Florham Park NJ 07932
wcohen@research.att.com

Abstract

Most databases contain "name constants" like course numbers, personal names, and place names that correspond to entities in the real world. Previous work in integration of heterogeneous databases has assumed that local name constants can be mapped into an appropriate global domain by normalization. Here we assume instead that the names are given in natural language text. We then propose a logic for database integration called WHIRL which reasons explicitly about the *similarity* of local names, as measured using the vector-space model commonly adopted in statistical information retrieval. An implemented data integration system based on WHIRL has been used to successfully integrate information from several dozen Web sites in two domains.

1 Introduction

Largely inspired by the proliferation of database-like sources on the World Wide Web, the integration of distributed, heterogeneous databases has become an active area of research (*e.g.*, [6; 9; 2; 8; 11; 1; 5; 13; 12]). Here we will describe a new approach to a little-studied aspect of this problem: *the integration of databases that lack common domains.*

In general, most databases contain many domains for which the individual constants correspond to entities in the real world; examples of such "name domains" include course numbers, personal names, company names, movie names, and place names. Most previous work in data integration either assumes these "name domains" to be global, or else assumes that local "name constants" can be mapped into a global domain by a relatively simple normalization process. However, examination of real-world information sources reveals many cases in which creating a global domain by normalization is difficult. For instance, in two Web databases listing educational software companies, we find the name constants "Microsoft" and "Microsoft Kids": do these denote the same company, or not?

In this paper, we reject the assumption that common domains can be easily constructed. Instead, we will assume that the *names assigned to real-world entities are given in natural language text.* We then propose a new logic for database integration called WHIRL. WHIRL is similar in flavor to probabilistic database systems (*e.g.*, [7]) in that it can draw tentative conclusions based on uncertain information. WHIRL retains the original local names and reasons explicitly about the *similarity* of pairs of names, using statistical measures of document similarity that have been developed in the information retrieval (IR) community.

This leads to a system that is in many ways an intermediate between statistical IR systems and conventional database systems. As in conventional database systems, the answer to a user's query is a set of tuples; however, as in many IR systems, these answer tuples are ordered so that the "best" answers are presented to the user first. In WHIRL, the tuples that are presented first are those for which the name co-reference conditions required by the user's query are considered most likely to hold.

In this paper, we describe an implemented data integration system based on WHIRL. This system has been used to integrate information from several dozen Web sites in two domains.

SIGMOD '98 Seattle, WA, USA

2 A Brief Description of WHIRL

As noted above, we will adopt a data model in which the names of real-world entities are represented as natural language text. One widely used method for representing text is the *vector space model* [10]. We assume a vocabulary T of *terms*, which will be treated as atomic; terms might include words, phrases, or word stems (morphologically derived word prefixes). A fragment of text is represented as *document vector*: a vector of real numbers $\mathbf{v} \in \mathcal{R}^{|T|}$, each component of which corresponds to a term $t \in T$. We will denote the component of $\mathbf{v}$ which corresponds to $t \in T$ by $\mathbf{v}^t$. The *similarity* of two document vectors $\mathbf{v}$ and $\mathbf{w}$ is given by the formula $sim(\mathbf{v}, \mathbf{w}) = \sum_{t \in T} \mathbf{v}^t \cdot \mathbf{w}^t$. If vectors are normalized to unit length, then $sim(\mathbf{v}, \mathbf{w})$ is always between zero and one.

The general idea behind this scheme is that the magnitude of the component $\mathbf{v}^t$ is related to the "importance" of the term t in the document represented by $\mathbf{v}$. Two documents are similar when they share many "important" terms. WHIRL adopts the widely TF-IDF weighting scheme, which assigns higher weights to terms that occur *infrequently* in the collection C. In a collection of company names, for instance, common terms like "Inc." and "Ltd." would have low weights; uniquely appearing terms like "Lucent" and "Microsoft" would have high weights; and terms of intermediate frequency like "Acme" and "American" would have intermediate weights.

WHIRL assumes that all data is stored in relations, but that the primitive elements of each relation are document vectors, rather than atoms. We call this data model STIR, for Simple Texts In Relations. Queries to this database are formulated in a query language that is conventional, save for

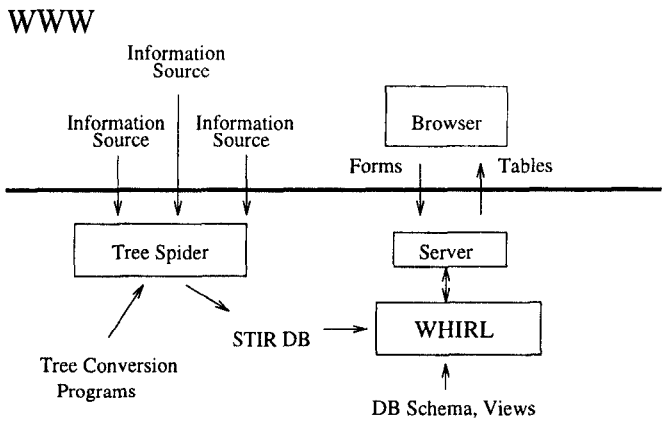

Figure 1: Architecture of the Information Access System

the addition of a built-in *similarity predicate*, written X~Y. As an example, the natural join of two predicates p and q containing different types of information about companies might be approximated by the query

`p(C1,Industry)` $\wedge$ `q(C2,WebSite)` $\wedge$ `C1~C2`

For semantics, define the *score* of a substitution θ for a conjunctive query Q defined to be zero if θ fails to satisfy some EDB literal in Q, and otherwise to be

$$score(\theta) = \prod_{\langle X_j, Y_j \rangle \in S_Q} sim(X_j\theta, Y_j\theta)$$

where S_Q is the set of pairs of variables that appear in the similarity literals of Q. Given a query, the WHIRL interpreter returns a list of the r highest-scoring substitutions $\theta_1, \ldots, \theta_r$, where r is a parameter set by the user. We call this list an *r-answer*.

Restricting the output of WHIRL to be r-answers is important, because in typical cases, the set of all substitutions with non-zero score will be huge. In the example above, for instance, any pair of company names C1, C2 that both contain the term "Inc" would lead to a non-zero score. WHIRL will not return all such pairings; instead it will return the r pairings for which C1 and C2 are most similar.

The full query language allows one to also express disjunctions of conjunctive queries; in this case similarity scores are combined as if they were independent probabilities (modulo certain necessary approximations.) If certain irredundancy assumptions are made, then WHIRL is a strict subset of Fuhr's probabilistic logic Datalog$_{PID}$ [7]. The novelty of WHIRL is mainly that the assumptions made in WHIRL enable relatively *efficient* inference, without making the logic too restricted to handle its intended task: integration of heterogeneous, autonomous databases by reasoning about the similarity of names. In particular, indexing and pruning methods from IR can be adapted to make inference in WHIRL quite fast, at least for moderate-sized databases.

A more detailed discussion of WHIRL can be found elsewhere [3; 4].

3 A Data Integration System based on WHIRL

We have implemented a system which uses WHIRL to integrate information from several Web sites. The architecture of the overall information system, which we call SPIRAL, is shown in Figure 1. SPIRAL departs from most data integration systems in collecting data off-line, rather than at query time. A spider program downloads Web pages and converts them, at download time, into STIR format; because it makes heavy use of HTML parse trees in this conversion step, this program is called a *tree spider* in the figure. The original HTML is then discarded, leaving a smaller STIR database that contains document vector representations of every simple text in the database, as well as a human-readable summary (currently, the first m characters) of every simple text. A *reference* for every simple text is also recorded for later use; the reference is the URL from which the text was obtained. The information retained by the spider is thus quite similar to the information stored by search engines like Altavista.

The STIR database is then loaded into WHIRL. Access to the data is provided by a server. The server accepts an HTTP request which encodes a conjunctive WHIRL query. This encoding is designed so that HTML forms can be conveniently used to generate queries. The server then executes the query with WHIRL, and returns an HTML table containing the r-answer to the query. Each field of this table corresponds to a simple text, and as in conventional search engines, this text is represented with a summary and a reference. Links associated with the table allow the user to request the next r substitutions.

The principle manual work involved in integration is programming the tree spider to convert pages. This process is only partly automated; with the current set of tools, each conversion routine takes around five minutes to develop and test. The programming language for describing HTML parse tree conversions is described in detail elsewhere [4]. In current prototype implementation, all document vector representations are stored in main memory, and document summaries and references are stored on disk.

To date two integration applications have been implemented with SPIRAL. One integrates information about North American birds from (at the time of this writing) 26 distinct sites. It includes geographic information (*e.g.*, a list of birds sighted in New Jersey), abundancy information (*e.g.*, lists of endangered birds), phylogenic information, and multimedia content (primarily bird calls, maps, and pictures). In all, more than 5800 distinct URLs are served. The second application integrates information about educational computer games for children. At the time of this writing, this application serves over 2700 distinct URLs from 16 sites, including over 2100 game reviews, as well as additional information such as recommendations, pointers to on-line demos, pricing information, and classifications of games by age group and subject matter.

Table 1 summarizes the runtime performance of WHIRL on some sample queries from the bird domain.[1] These queries were generated by HTML forms, which allow the user to state questions in certain restricted formats, and have been "translated" by showing the English text appearing on the form after the user's selections have been made. For each query, the top 20 answers were retrieved. Clearly, WHIRL is quite efficient, even when computing three- and four-way joins.

[1] On an SGI Challenge with 250 MHz R10000 processors.

Time (sec)	Query
0.04	na(Ord,N) ∧ Ord∼"passeriforms" ∧ es(ESN,Stat) ∧ ESN∼N
	"tell me the names of birds in the order passeriforms that have been sighted in North America and are endangered or threatened"
0.08	na(Ord,N) ∧ smap(MN,HRef) ∧ MN∼N ∧ nj(NJN) ∧ NJN∼N ∧ w(WN) ∧ WN∼N
	"show me the summer range of birds that have been sighted in New Jersey and are the Audubon Society's watchlist"
0.02	na(Ord,N) ∧ Ord∼"pelicaniforms" ∧ img(IN,HRef) ∧ IN∼N ∧ nj(NJN) ∧ NJN∼N ∧ es(ESN,Stat) ∧ ESN∼N
	"show me pictures of birds in the order pelicaniforms that have been sighted in New Jersey and are endangered or threatened"

Table 1: Performance of the WHIRL interpreter on sample queries. Predicate names have been abbreviated for formatting reasons.

4 Conclusions

SPIRAL demonstrates a number of important points about the textual, similarity-based approach to integration adopted by WHIRL. From a human factors viewpoint, it is relatively easy to users to express complex queries that integrate information from several sources, using an appropriate set of HTML forms. Furthermore, the interaction with the database system is fairly natural, due to its similarity to Internet search engines. From the viewpoint of an implementor or maintainer of an integration system, there are also several advantages. Integration is facilitated by the fact that normalization of constants is not required, and that (due to the flexibility afforded by the similarity-based reasoning) extraction of data items from the original information sources can be only approximate. Another advantage is that queries are executed against a locally-stored index, rather than being executed by submitting subqueries to external information sources. This means that if one or more information sources change format or become temporarily unavailable, the impact on the end user is minimal; queries can still be executed quickly, although the answers may contain some stale URLs.

A final point demonstrated by SPIRAL is that the approach to integration represented by WHIRL is scalable to usefully large problems. In particular it is possible to integrate large segments of the Web, while maintaining fast response times.

References

[1] Serge Abiteboul and Victor Vianu. Regular path queries with constraints. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-97)*, Tucson, AZ, May 1997.

[2] Yigal Arens, Craig A. Knoblock, and Chun-Nan Hsu. Query processing in the SIMS information mediator. In Austin Tate, editor, *Advanced Planning Technology*. AAAI Press, Menlo Park, CA, 1996.

[3] William W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. In *Proceedings of ACM SIGMOD-98*, Seattle, WA, 1998.

[4] William W. Cohen. A Web-based information system that reasons with structured collections of text. In *Proceedings of Autonomous Agents-98*, St. Paul, MN, 1998.

[5] Oliver M. Duschka and Michael R. Genesereth. Answering recursive queries using views. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-97)*, Tucson, AZ, May 1997.

[6] Oliver M. Duschka and Michael R. Genesereth. Query planning in infomaster. In *Proceedings of the Twelfth Annual ACM Symposium on Applied Computing (SAC97)*, San Jose, CA, February 1997.

[7] Norbert Fuhr. Probabilistic Datalog—a logic for powerful retrieval methods. In *Proceedings of the 1995 ACM SIGIR conference on research in information retrieval*, pages 282–290, New York, 1995.

[8] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. The TSIMMIS approach to mediation: Data models and languages (extended abstract). In *Next Generation Information Technologies and Systems (NGITS-95)*, Naharia, Israel, November 1995.

[9] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the 22nd International Conference on Very Large Databases (VLDB-96)*, Bombay, India, September 1996.

[10] Gerard Salton, editor. *Automatic Text Processing*. Addison Welsley, Reading, Massachusetts, 1989.

[11] Dan Suciu. Query decomposition and view maintanance for query languages for unstructured data. In *Proceedings of the 22nd International Conference on Very Large Databases (VLDB-96)*, Bombay, India, 1996.

[12] Dan Suciu, editor. *Proceedings of the Workshop on Management of Semistructured Data*. Available on-line from http://www.research.att.com/ suciu/workshop-papers.html, Tucson, Arizona, May 1997.

[13] Anthony Tomasic, Remy Amouroux, Philippe Bonnet, and Olga Kapitskaia. The distributed information search component (Disco) and the World Wide Web. In *Proceedings of the 1997 ACM SIGMOD*, May 1997.