WebML: Querying the World-Wide Web for Resources and Knowledge

Osmar R. Zaïane Jiawei Han
School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6
{zaiane, han}@cs.sfu.ca

Abstract

There is a massive increase of information available on electronic networks. This profusion of resources on the World-Wide Web gave rise to considerable interest in the research community. Traditional information retrieval techniques have been applied to the document collection on the Internet, and a myriad of search engines and tools have been proposed and implemented. However, the effectiveness of these tools is not satisfactory. None of them is capable of discovering knowledge from the Internet. We propose a declarative query language that would allow resource discovery on the Internet with interactive and progressively refined inquiries. The language also consents to the discovery of knowledge within the content of the documents and the structure of the hyperspace.

1 Introduction

More than half a century ago, in a paper in which he describes the "Memex", a system for storing and organizing multimedia information, Vannevar Bush invited researchers to join the effort in building an information system for holding the human knowledge, and making it easily accessible[4]. He writes: "A record, if it is to be useful... must be continuously extended, it must be stored, and above all it must be consulted." A massive aggregation of documents is now stored on the Internet. The World-Wide Web is holding a colossal collection of resources, from structured records, images and programs to semi-structured files and free text documents. The availability of information is not questionable. We are actually overwhelmed by this excess of information. Accessibility as described by Vannevar Bush however is still unsolved. For many decades, information retrieval from document repositories has drawn much attention in the research community. Many techniques have been proposed and implemented in successful and less prevailing applications. With the advent of the World-Wide Web, the appearance of a panoply of services and accumulation of a colossal aggregate of resources, information retrieval techniques have been adapted to the Internet, bringing forth indexing models and search engines. However, the effectiveness of these tools is not satisfactory and is even irritating. The annoying results of current search engine technologies have invited researchers to tackle new challenges. Better indexing approaches, specialized information gathering agents, filtering and clustering methods, etc. have since been proposed.

A new research trend in the field of information retrieval from the World-Wide Web is web querying and the design of query languages for semi-structured data. The approach for querying structured and semi-structured documents involves the construction of tailored wrappers that map document features into instances in internal data models (i.e. graphs or tables). The introduction of new types of documents usually necessitates the construction of new custom-made wrappers to handle them. Due to the semi-structured nature of web pages written in HTML, the migration of semi-structured data query languages like UnQL[3] and Lorel[1] to the World-Wide Web domain is evident. W3QL[7], WebLog[8], WebSQL[9] and WebOQL[2] are all intended for information gathering from the World-Wide Web. While WebLog and WebOQL aim at restructuring web documents using Datalog-like rules or graph tree representations, WebSQL and W3QL are languages for finding relevant documents retrieved by several search engines in parallel. However, none of these approaches takes advantage of the structure of the global information network as a whole. Moreover, none of these languages performs data mining from the Web. A language like WebSQL is built on top of already existing search engines which lack precision and recall. W3QS, the system using W3QL, also uses existing search engines. A web document structuring language like WebOQL or WebLog is capable of retrieving information from on-line news sites like CNN, tourist guides, or conference lists, but is limited to a subset of the web defined in the queries. Their powerful expressions, however, can extract intersting and useful information from within a given set of web pages. We intend to use this power to build our system's data model. We propose a web query language, WebML that permits resource discovery as well as knowledge discovery from a subset of the Internet or the Internet as a whole. WebML is an SQL-like declarative language for web mining. We have introduced new primitives that we believe make the language simple enough for casual users. These primitives allow powerful interactive querying with an OLAP (OnLine Analytical Processing)-like interaction (i.e. drill-down, roll-up, slice, dice, etc.). The language takes advantage of a Multi-Layered DataBase (MLDB) model[5, 12] in which each layer is obtained by successive transformations and generalizations on lower layers, the first layer being the primitive data from the Internet. The higher strata are stored in relational tables and take advantage of the relational database technology. Their construction is based on a propagation algorithm and assumes the presence or availability of descriptive metadata, either provided by document authors or extracted by tools like WebLog and WebOQL.

The remainder of the paper is organized as follows: In section 2 the MLDB model and its construction are introduced. Section 3 presents the WebML query language. Some query examples are discussed in Section 4. Finally, in section 5, we present our conclusion and possible directions for future work.

2 Multi-Layered Database Model

The philosophy behind the construction of MLDB is information abstraction, which assumes that most users may not like to read the details of large pieces of information but may like to scan the general description of the information. Our motivation is not web page restructuring, like with WebLog and WebOQL, but rather web page content and web page inter-relations abstraction.

A multiple layered database (MLDB) consists of 3 major components: a database schema, which contains the meta-information about the layered database structures, a set of concept hierarchies, and a set of (generalized) database relations at the nonprimitive layers of the MLDB and files in the primitive global information base.

The first component, a database schema, outlines the overall database structure of the MLDB. It stores general information such as structures, types, ranges, and data statistics about the relations at different layers, their relationships, and their associated attributes as well as the location where the layers reside. Moreover, it describes which higher-layer relation is generalized from which lower-layer relation and how the generalization is performed. Therefore, it presents a route map for data and meta-data browsing.

The second component, a set of concept hierarchies, provides a set of predefined concept hierarchies which assist the system to generalize lower layer information to high layer ones and map queries to appropriate concept layers for processing. These hierarchies are also used for query-less browsing of resources like drill-down and roll-up operations.

The third component consists of the information base at the primitive information level (i.e., $layer_0$) and the generalized database relations at the nonprimitive layers.

The third component is by definition dynamic. The schema defined in the first component of the MLDB model can also be enriched with new fields, and new route maps can be defined after the system has been initially conceived. The updates are incremental and are propagated, in the case of the schema update, from lower layers to higher ones. New concept hierarchies can be defined as well, or updated. While updates to current concept hierarchies imply incremental updates in layered structure, new concept hierarchies may suggest the definition of a new set of layers or an analogue MLDB.

Because of the diversity of information stored in the global information base, it is difficult, and even not realistic, to create relational database structures for the primitive layer information base. However, it is possible to create relational structures to store reasonably structured information generalized from primitive layer information.

For example, based on the accessing patterns and accessing frequency of the information base, $layer_1$ is organized into dozens of database relations, such as *document, person, organization, software, map, library_catalog, commercial data, geographic_data, scientific_data, game*, etc. The relationships among these relations can also be constructed either explicitly by creating relationship relations as in an entity-relationship model, such as *person-organization*, or implicitly (and more desirably) by adding the linkages in the tuples of each (entity) relation during the formation of $layer_1$, such as adding URLs (Uniform Resource Locator).

Notice that an incremental updating of the schema, such as adding new attributes at $layer_1$, may imply incremental updating and propagating the lower layer information to higher ones in the multiple-layered database, which may also require incremental updates of the layer building softwares.

2.1 Construction of the MLDB structure

The goal for the construction of the MLDB is to transform and/or generalize the unstructured data of the primitive layer at each site into relatively structured data, manageable and retrievable by the database technology.

Specialized tools, similar to Essence[6] are executed locally on information provider sites to extract pertinent data from documents. WebLog and WebOQL-like query languages can also be exploited to gather the needed information from within documents. This information is stored in the first layer and is generalized in higher levels. The $layer_1$ is distributed and resides locally on each information provider site. It is only the higher levels that are gathered in a centralized location and mirrored for better performance.

Extracting information from structured bibtex files or postscript papers is fairly smooth. However, most web pages don't easily convey the needed information. The extensible markup language (XML) developed by the World-Wide Web Consortium will help in this direction. Many web publishing tools are adopting XML and will help promote widespread improved structured web documents. The Dublin Metadata workshop has stressed the importance of metadata (i.e. document descriptors) in networked documents to facilitate resource discovery [10]. Already, extensions to the HTML specifications include some tags allowing the description of keywords and content summary inside the HTML document. Because of their use by search engines in their ranking of documents, more web document authors are now willing to manually add these descriptions in their web pages.

To simplify our discussion, we assume in this paper that the $layer_1$ database contains only two extended relations, document and person. Other relations can be constructed and generalized similarly.

1. document(*file_addr, authors, title, publication, pub_date, abstract, language, table_of_contents, category_description, keywords, index, URLs, multimedia_attached, num_pages, form, size_doc, time_stamp, access_frequency, ...*).
2. person(*last_name, first_name, home_page_addr, position, picture_attached, phone, e-mail, office_address, education, research_interests, publications, size_of_home_page, timestamp, access_frequency, ...*).

Take the *document* relation as an example. Each tuple in the relation is an abstraction of one *document* from the information base. The whole relation is a detailed abstraction (or descriptor) of the information in documents gathered from a site. The relations in $layer_1$ are substantially smaller than the primitive layer from the information base, but are still rich enough to preserve most of the interesting pieces of general information for a diverse community of users to browse and query. The two $layer_1$ relations, document and person, are further generalized into $layer_2$ database. The resulting relations are usually smaller with less attributes and records. Least popular fields from $layer_1$ are dropped, while the remaining fields are inherited by the $layer_2$ relations. Relations are split according to different classification schemes, while tuples are merged relying on successive subsumptions according to the concept hierarchies used. General concept hierarchies are provided explicitly by domain experts. Other hierarchies are built automatically and stored implicitly in the database. We have implemented and will propose in a forthcoming paper a technique for the construction of a concept hierarchy for keywords extracted from web pages using an enriched WordNet semantic network[11].

Due to lack of space, we do not discuss the MLDB construction and the generalization problem further, but refer the reader to [5, 12] for more details.

3 Web Mining Language

Similar to other extended-relational database systems, a MLDB system treats the requests for information browsing and resource discovery like relational queries. However, since concepts in a MLDB are generalized at different layers, search conditions in a query may not match exactly the concept level of the currently inquired or available layer of the database. For example, to find documents related to a particular topic, such as "attribute-oriented induction", a query may put this term as a search key. However, the current layer may only contain terms corresponding to a higher concept level, such as "induction techniques", or "data mining methods". In this case, it is unlikely to find in the current layer an exact match with the provided search key, but is likely to find a more general concept that absorbs the search key. On the other hand, a search key in a query may be at a more general concept level than those at the current layer. For example, a search key "sports", though conceptually covers the term "baseball", does not match it in the database. Therefore, a key-oriented search in an MLDB leads us to introduce several additional relational operations to extend the semantics of traditional selection and join. Four relationships, *coverage, subsumption, synonymy,* and *approximation.* These operators have their correspondent language primitive in WebML defined respectively as covers, coveredby, like and closeto.

<WebML> ::= <MINE HEADER> **from** relation_list
  [**related-to** name_list] [**in** location_list]
  where where_clause
<MINE HEADER>::=
  {{ **select** | **list** } { attributes_name_list | $*$ }
  | <DESCRIBE HEADER> | <CLASSIFY HEADER>}
<DESCRIBE HEADER>::= **mine description**
  in-relevance-to { attributes_name_list | $*$ }
<CLASSIFY HEADER>::= **mine classification**
  according-to attributes_name_list
  in-relevance-to { attributes_name_list | $*$ }

Table 1: The top level syntax of WebML.

The top-level WebML query syntax is presented in Table 1. At the position for the keyword select in SQL, an alternative keyword list can be used when the search is to browse the summaries at a high layer, mine description can be used when the search is to discover and describe the general characteristics of the data, mine classification is used to find classifications of web objects according to some attributes, whereas select remains to be a keyword indicating to find more detailed information. Two optional phrases, "related-to *name_list*" and "in *location_list*", are introduced in WebML for quickly locating the related subject fields and/or geographical regions (e.g., Canada, Europe, etc.). They are semantically equivalent to some phrases in the where-clause, such as "*keyword* covered-by *field_names*" and/or "*location* covered-by *geo_areas*", etc. But their inclusion not only makes the query more readable, but also helps the system locate the corresponding high layer relation if there exists one. The phrase "according-to *attributes_name_list* in-relevance-to attributes_name_list" is only used for classification with mine classification. It indicates the attributes upon which to classify web objects. The where-clause is similar to that in SQL except that new operators may be used.

While this query language is simple, users do not have to learn it and write queries. A Java-based or HTML-based user interface can easily be developed on top of WebML to avoid heavy instruction queries, and to provide a means for interaction based on field-filling and button-clicking. This is one of our future projects.

4 WebML Examples

As mentioned earlier in this paper, the MLDB structure provides ground for resource discovery on the Internet (i.e. pinpointing relevant documents) as well as knowledge discovery (i.e. implicit knowledge extraction). Following are examples of queries for resource discovery and for data mining from the Web.

Example 4.1 The query, *list the documents published in Europe and related to "data mining"*, is presented as follows.

list $*$ **from** document **in** Europe
related-to computing science
where **one of** keywords **covered-by** "data mining"

Notice that the keyword list indicates that the query is to briefly browse the information, and therefore, it searches the relations using the where-clause as a constraint. Using select instead of list would locate a set of URL addresses of the required documents, together with the important attributes of the documents. The keyword list, however, allows to display document attributes at a high conceptual level and provides and OLAP-like interaction. "from *document*" does not indicate to find the document relation at $layer_0$ or $layer_1$, but indicates to find the top-most layer of the *document* relation which fits the query. Therefore, "*document*" is a clue to the system to find the appropriate relation at a high layer. We adopt this convention since it is the system's responsibility to find the best match, and it is unreasonable to ask users to remember all the relation names at different layers. Moreover, the related-to clause can help the system locate the appropriate top layer relation in case the relations are split by topic. To execute this query, the MLDB system uses the phrase "from *document*" and "related-to *computing science*" to locate the top layer relation, *cs_document* for example. The phrase, "one of *keywords* covered-by *'data mining'*" means that there exists an entry in the set *keywords* which is subsumed under '*data mining*'. Moreover, the phrase "in *Europe*" confines the search to be within *Europe* which will be mapped into concrete countries using a concept hierarchy for Internet domains. In this case, a relatively large set of answers will be returned. An interactive process to deepen the search will usually be initiated by users after browsing the answer set.

Example 4.2 To inquire about European universities productive in publishing on-line popular documents related to database systems since 1990, a WebML query is presented as follows:

select affiliation **from** document **in** Europe
where affiliation **belong-to** "university" **and**
  one of keywords **covered-by** "database systems"
  and publication_year $>$ 1990 **and** count $>$ 20
  and access_frequency **belong-to** "high"

In this query, "productive" is measured as those containing more than 20 published papers on the Internet since 1990 related to database systems, which is either be obtained based on the result of an initial browsing of the *document* table, or justified by some interactive queries. The term "high" is a generalization of the numeric value of access_frequency along its concept hierarchy. What is interesting to note is that the execution of this query does not return a list of document references, but rather a list of universities (publishing popular documents about databases), which is knowledge extracted from a conglomerate of documents.

Example 4.3 Suppose the query is to "*describe the general characteristics in relevance to authors' affiliations, publications, etc. for those documents which are popular on the Internet and are on "data mining"*. A knowledge discovery query to answer this request, characterized by the keyword "describe" as shown below.

mine description
in-relevance-to authors.affiliation, publication, pub_date
from document **related-to** Computing Science
where **one of** keywords **like** "data mining"
and access_frequency = "high"

The discovery query will be first executed as a retrieval to collect from *cs_document* the data which are relevant to "*authors.affiliation, publication, pub_date*" and satisfy the where-clause. Then the attribute-oriented induction is performed on the collected data, which generalizes "*publication*" into groups, such as major AI journals, major database conferences, and so on, and generalizes publication date to year, etc. The generalized results are collected in a datacube and can be interactively manipulated by the user using OLAP operations.

Example 4.4 To classify according to update time and popularity the documents published on-line in sites in the canadian and commercial internet domain after 1993 and about information retrieval from the Internet, a WebML query can be presented as follows:

mine classification
according-to timestamp, access_frequency
in-relevance-to *
from document **in** Canada, Commercial
where **one of** keywords **covered-by** "information retrieval"
and one of keywords **like** "Internet"
and publication_year > 1993

The phrase mine classification requests a classification tree from the system. The query first collects the relevant set of data from the MLDB relations, executes a data classification algorithm to classify documents according to their access frequency and their last modification date, then presents each class and its associated characteristics in a tree. The user can navigate the tree representation and drill through to the documents if needed.

5 Conclusion and Future Work

Search engines currently available on the Internet are keyword-driven, and the answers presented are lists of presumably relevant documents. The MLDB and WebML allow us to apprehend and solve the resource discovery issues by presenting lists of relevant documents to users, but also allowing the users to progressively and interactively browse detailed information leading to a targeted set of pertinent documents. WebML queries are treated like information probes, being mapped to a relatively high concept layer and answered in a hierarchical manner. Moreover, the knowledge discovery power of WebML is unique. It helps find interesting high level information about the global information base. It provides users with a high-level view of the database, statistical information relevant to the answer set, and other associative and summary information at different layers. In addition, the MLDB model takes advantage of web page restructuring query languages and available networked agents to retrieve pertinent descriptors from web documents.

Experiments run locally on a collection of on-line documents were very promising. We plan to extend these experiments and include full operational web sites. The design and implementation of a point-and-click user interface is under way. The interface will alleviate the need for writing queries directly in WebML, and it will also allow interactive OLAP on a hyperspace datacube.

References

[1] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener. The lorel query language for semistructured data, 1997. http://www-db.stanford.edu/~abitebou/pub/jodl97.lorel96.ps.

[2] G. O. Arocena and A. O. Mendelzon. WebOQL: Restructuring documents, databases and webs. In *Proc of ICDE Conf.*, Orlando, Florida, USA, February 1998.

[3] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In *Proc. ACM SIGMOD Conf. on Management of Data*, 1996.

[4] Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[5] J. Han, O. R. Zaïane, and Y. Fu. Resource and knowledge discovery in global information systems: A scalable multiple layered database approach. In *In Proc. Conf. on Advances in Digital Libraries*, Washington, DC, May 1995.

[6] D. Hardy and M. F. Schwartz. Essence: A resource discovery system based on semantic file indexing. In *Proc. of the USENIX Winter Conf.*, Berkeley, CA, 1993.

[7] D. Konopnicki and O. Shmueli. W3QS: A query system for the world-wide web. In *Proc. 21st VLDB Conf.*, Zurich,Switzerland, 1995.

[8] L. Lakshmanan, F. Sadri, and I. Subramanian. A declarative language for querying and restructuring the web. In *Proc. 6th Int. Workshop on Research Issues in data Engineering*, New Orleans, 1996.

[9] Alberto Mendelzon, George Mihaila, and Tova Milo. Querying the world wide web. In *Proc. PDIS'96*, Miami, December 1996.

[10] S. Weibel, J. Godly, E. Miller, and R. Daniel. OCLC/NCSA metadata workshop report, March 1995.

[11] WordNet - a lexical database for english. http://www.cogsci.princeton.edu/~wn/, 1998.

[12] O. R. Zaïane and J. Han. Resource and knowledge discovery in global information systems: A preliminary design and experiment. In *Proc. Conf. On Knowledge Discovery and Data Mining*, Montreal, Canada, 1995.