Semiautomatic Generation of Data-Extraction Ontologies

A Thesis Proposal Presented to the
Department of Computer Science
Brigham Young University

In Partial Fulfillment of the Requirements
for the Degree of Master of Science

Yihong Ding

July 3, 2001

I. Introduction

The amount of data available on the World Wide Web has been growing explosively during the past few years. As the Web has grown, people have become more and more interested in obtaining user-specified information from this fast-growing body of knowledge similar to the way we can query a database for user-specified information.

Traditionally, two approaches have been taken to enable users to query the Web like a database. One is to enhance traditional query languages to make them "Web aware" so that data in Web pages can be queried directly (e.g., [AM98] [MMM97]). The more popular approach, however, is to either actually or virtually extract the information contained within web pages by wrappers (e.g., [AK97] [LCC99]). Once extracted, standard query languages, such as SQL, can be applied.

Wrappers, however, typically very much depend on the structure of the source documents. Therefore, it is difficult to apply these source-dependent wrappers to the documents with different formats, even when the documents contain the same or similar information. As a solution to this problem, the data extraction group ([DEG]) at Brigham Young University has defined an approach based on data-extraction ontologies that extracts data from data-rich, unstructured, multiple-record Web documents. The data-extraction ontology is source-independent and can extract data with high precision and recall without change from documents having the same type of information, but in different formats ([CDE+00] [ECS+98] [ECJ+98] [ECJ+99] [ENX01]).

A data-extraction ontology for an application is defined based on the OSM model ([EKW92] [Emb98] [Lid95]), which consists of lexical and non-lexical object as well as the relationship and constraints among them. For each lexical object set, a data frame ([Emb80]) defines the appearance of lexical objects. Once a data-extraction ontology has been defined for an application, the system parses the application ontology to generate a database scheme and to generate matching rules for constants and keywords. Then a record extractor extracts the data based on the generated rules and schema and stores the result in a relational database. Performance analysis for several application ontologies shows that for most of the applications, both the recall ratios and the precision ratios are over 90% ([ECJ+99]).

Unfortunately, widespread usage of this extraction ontology approach is restricted because of the expense related to the time required to manually discover and create the ontology. Moreover, despite the fact that many people would like to create applications of personal interest, it is normally impossible for an untrained user to create an application ontology because of the knowledge and skill required to create a custom-built ontology. Therefore, a critical problem for this extraction ontology approach is to enable users to rapidly create application ontologies. Thus, the focus of this research is to solve this problem, i.e., to, at least, semi-automatically generate an application ontology based on available knowledge resources and sample input documents.

II. Thesis Statement

This research focuses on enabling semiautomatic creation of data-extraction ontologies based on existing knowledge bases and element recognizers, along with some construction heuristics and some input Web source documents.

III. Methods

This research is to be done in three steps. First, gather the necessary knowledge and transform it into a useable form. Second, automatically generate an initial data-extraction ontology based on the acquired knowledge and sample target documents. Third, provide a way to let users evaluate the performance of the generated ontology and refine it, if necessary. Further, to evaluate the success of the research, we measure how well the tool performs with respect to how well it could have performed.

Gathering Knowledge

In order to build a data-extraction ontology, we must first have the knowledge. Sources of knowledge could be anything that contains the knowledge of interest, such as a list of names, a regular expression for some special string like a date, a traditional relational database, a general ontology like Mikrokosmos ([Mik]), or even some other knowledge collection like an encyclopedia. A problem, however, is that these knowledge sources may be stored in totally different formats so that it is very hard to handle them all.

XML appears to be the best choice for our knowledge storage format. The schema, relations, constraints, and records of the OSM-based data-extraction ontology can be mapped to and from XML. Furthermore, XML is being used more and more for data exchange on the Web ([XmlA] [XmlI]). Thus, it is becoming easier to find the knowledge of interest in XML ([Bou99], [Bou00]), and easier to find tools to transfer knowledge to XML ([Bou00(2)]).

Another knowledge-gathering issue is the question of how to deal with special strings, such as dates, phone numbers, etc., for which we must provide regular expressions. Because it is very hard to generate these recognizers automatically, we may have to create this part of the knowledge manually, i.e., we will provide a standard string recognizer library. Fortunately, there are some already developed regular expressions that can be used directly in the recognizer library (e.g., [KCG+96] [ECJ+99] [Lyo00]); hopefully we can find some others.

Generating the Initial Ontology

Once enough knowledge has been gathered and converted to XML, we can collect the various XML knowledge documents together to build a high-level schema. This high-level schema defines the set of attributes that may appear in a generated data-extraction ontology.

Every attribute can be classified into one of the two categories: domain-specific or domain-independent. Attributes, like country name and car model, have a list of strings, each of which represents an object instance. These attributes are categorized as domain-specific attributes. The high-level schema associates these attributes directly with their instances. Other attributes, for example year, date, and population, are different from domain-specific attributes because these values may play different roles in different domains (e.g., event dates in history and due dates for school projects). For this type of attribute we are interested in regular expressions for values and for the context related to the occurrences of these values. Whereas domain-specific values usually associate with a single attribute in the high-level schema, domain-independent values often associate with several attributes. Note that since a list of strings can also be thought of as a regular expression, both categories have associated regular expressions.

After a high-level schema has been generated and value sets are associated with attributes, we can start to select the appropriate attributes for the application based on recognized values in an input training document. The selection process will be done in two steps: initial recognition and ambiguity elimination. For both domain-specific and domain-independent attributes, we will apply regular-expression recognizers in an attempt to recognize information in the training document ([BM98] [ECJ+99] [YL99]). If a recognizer finds some data item, each of the attributes with which the recognizer associates is a potential attribute for the data item for the generated ontology.

Two sorts of ambiguities will exist for potential matched attributes. One ambiguity occurs because of different specializations for the same generalization, e.g., check-in date vs. check-out date or import products vs. export products. Usually these ambiguous values are recognized by same regular expressions. To remove this sort of ambiguity, we can look at the context of each occurrence and use WordNet ([WN]) and other dictionaries to eliminate the ambiguity.

The other sort of ambiguity occurs because different recognizers may recognize the same string in the training document. If the ambiguous attributes are closely related in the high-level schema, the same context identification process as discussed above must be applied. More often, however, the attributes will be scattered in different places in the high-level schema. Therefore, a simpler way to solve this ambiguity is to look for a cluster of potential matches and consider all potential matches outside this cluster to be errors and eliminate them.

After resolving the ambiguities and finding a cluster of potential matches, it is straightforward to generate the initial data-extraction ontology. The matched attributes within the cluster and their relationships define the structure for the extraction ontology, and their respecting regular expressions become the regular expressions for the data frames in the extraction ontology.

User Validation

Once an initial data-extraction ontology has been generated, a user can check it by applying it to a set of validation documents using the ontology extraction tool ([DEG]). If the results are not satisfactory, a user can apply the OntologEditor ([Hew00]) to the generated ontology. The OntologEditor provides a method of editing an Object Relationship Model (ORM) and its associated data frames and also provides debugging functionality for editing regular expressions in data frames by displaying sample text with highlighting on sample source documents. As an alternative to refining the generated data-extraction ontology with the OntologyEditor, we may consider allowing a user to view and verify the knowledge sources directly so that a better data-extraction ontology can be generated.

Ontology Generation Performance Evaluation

To evaluate the performance of the data-extraction ontology generation process, we can apply three types of evaluations.

1. We can measure how much of the data-extraction ontology was generated with respect to how much could have been generated.
2. We can measure the amount of components generated that should not have been generated.

3. We can measure precision and recall for each lexical object set in generated extraction ontology.

IV. Contribution to Computer Science

This research will provide a way to semiautomatically generate a data-extraction ontology. To achieve this, we will show how to exploit existing knowledge, and we will update and link the currently available data-extraction tools. Also, a string recognizer ontology library will be provided to extract numbers and other special strings.

V. Delimitations of the Thesis

This research will not attempt to do the following:

- Use all possible storage formats of existing knowledge. The input knowledge base will be restricted to XML, or to formats easily transformed to XML.
- Handle documents other than HTML or plain text documents written in English.
- Update the input knowledge source. It is assumed that the knowledge source will be sufficient to cover all the necessary requirements for the specific applications of interest.

VI. Thesis Outline

1. Introduction (3 pages)
2. Related Work (2 pages)
3. Input Resources (7 pages)

   3.1. Domain-Specific Knowledge Bases Preparing

   3.2. String Recognizers

   3.3. Training Documents
4. Methodology (20 pages)

   4.1. Architecture

   4.2. High-Level Schema

   4.3. Initial Data-Extraction Ontology Generation

   4.4. User Interface

   4.5. Update Strategy
5. Experimental Analysis and Results (10 pages)
6. Conclusions, Limitations, and Future Work (4 pages)

VII. Thesis Schedule

A tentative schedule of this thesis is as follows:

Literature Search and Reading	January – July 2001
Chapter 3	April – October 2001
Chapter 4	April – November 2001
Chapters 1 and 2	October 2001 – November 2001
Chapters 5 and 6	November – December 2001
Thesis Revision and Defense	December 2001

VIII. Bibliography

[AK97] N. Ashish and C. Knoblock, Wrapper Generation for Semi-structured Internet Sources, *SIGMOD Record*, Vol. 26, No. 4, pp. 8-15, December 1997.

> This paper presents an approach for semi-automatically generating wrappers for querying semi-structured WWW sources.

[AM98] G.O. Arocena and A.O. Mendelzon, WebOQL: Restructuring documents, databases and webs, In *Proceedings of the 14th International Conference on Data Engineering*, pp. 24-33, Orlando, Florida, February 1998.

> This paper presents the WebOQL system that supports a general class of data restructuring operations in the context for the Web. WebOQL synthesizes ideas from query languages for the Web, for semi-structured data and for website restructuring.

[BM98] L.D. Baker and A.K. McCallum, Distributional clustering of words for text categorization, In *Proceedings of the 21st Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 96-103, Melbourne, Australia, August 1998.

> This paper describes the application of Distributional Clustering to document classification.

[Bou99] R. Bourret, XML and Databases, 1999. http://www.rpbourret.com/xml/XMLAndDatabases.htm#isxmladatabase

> This linked paper discusses the relations between XML and databases. It was last updated on November 2000.

[Bou00] R. Bourret, XML Database Products, 2000. http://www.rpbourret.com/xml/XMLDatabaseProds.htm#xmlanddatabases

> This link gives a list of available software for XML database products. Though it may not be complete, it is very helpful to know the progress of XML database products. It was last updated on May 22, 2001.

[Bou00(2)] R. Bourret, Data Transfer Strategies: Transferring data between XML documents and relational databases, 2000. http://www.rpbourret.com/xml/DataTransfer.htm

This linked paper discusses strategies for transferring data between XML documents and relational databases according to two mappings (a table-based mapping and an object-based mapping) commonly used to map DTDs to relational databases.

[CDE+00] D.M. Campbell, Y. Ding, D.W. Embley, K.Hewett, D.L. Jackman, S.S. Jeffires, Y.S. Jiang, D. Lewis, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, A.L. Peacock, D.J. Seer, R.D. Smith, S.H. Yau, M. Xu, and L. Xu, Demonstration: A Robust Web Data-Extraction Technique With High Recall and Precision, *DEG Tech. Report*, Provo, Utah, 2000.

This report describes a demo to show how to extract and structure data found in data-rich, unstructured, multiple-record Web documents.

[DEG] DEG group and ontology demo home page:
http://www.deg.byu.edu/

This web site gives information about Data Extraction Research Group at Brigham Young University, and also the Web demo for the application ontology.

[ECS+98] D.W. Embley, D.M. Campbell, R.D. Smith, and S.W. Liddle, Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents, In *Proceedings of 7th International Conference on Information and Knowledge Management* (*CIKM'98)*, pp. 52-59, Washington, D.C., November 1998.

This paper presents the ontology approach to extracting information from unstructured documents. It discusses the detailed structure of the ontology components.

[ECJ+98] D.W. Embley, D.M. Campbell, Y.S. Jiang, Y.-K. Ng, and R.D. Smith, A Conceptual-Modeling Approach to Extracting Data from the Web, In *Proceedings of 17th International Conference on Conceptual Modeling (ER'98)*, pp. 78-91, Singapore, November 1998.

This paper presents the ontology as a conceptual-modeling approach. By parsing the ontology, it can automatically produce a database scheme and recognizers for constants and keywords, and then invoke routines to recognize and extract data from unstructured documents and structure it according to the generated database scheme.

[ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith, Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages, *Data and Knowledge Engineering*, Vol. 31, No. 3, pp. 227-251, 1999.

This paper presents the definition of the data-extraction ontology and the way to extract data from the Web documents through data-extraction ontology. Their experiments show that it is possible to achieve good recall and precision ratios for documents that are rich in recognizable constants and narrow in ontological breath.

[EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield, *Object-oriented Systems Analysis: A Model-Driven Approach,* Prentice Hall, Englewood Cliffs, New Jersey, 1992.

This book describes an approach to understanding how a software system, such as a traditional database or a real-time scientific system, should work, before you begin designing it with your favorite design technique and programming language.

[Emb80] D.W. Embley, Programming with Data Frames for Everyday Data Items, In *Proceedings of AFIPS National Computer Conference (NCC'80)*, Vol. 49, pp. 301-305, Anaheim, California, May 1998.

This paper describes the definition of data frames.

[Emb98] D.W. Embley, *Object Database Development: Concepts and Principles,* Addison-Wesley, Reading, Massachusetts, 1998.

This book presents the fundamental principles and concepts needed for developing advanced database applications, and shows how to apply these principles successfully.

[ENX01] D.W. Embley, Y.-K. Ng, and Li Xu, Recognizing Ontology-Applicable Multiple-Record Web Documents, In *Proceedings of 20th International Conference on Conceptual Modeling (ER'2001),* Yokohama, Japan, November 2001 (to appear).

This paper proposes a technique for recognizing multiple-record Web documents apply to an ontologically specified application. Combining machine-learned rules defined over three heuristic measurements, the authors determine whether a Web document is applicable for a given ontology.

[FB] The world factbook home page:
http://www.odci.gov/cia/publications/factbook/

This web site gives information about the geographic, economical, governmental, etc. information of all the countries over the world. It is owned by the Central Intelligence Agency (CIA) in USA and is updated each year.

[Hew00] K.A. Hewett, *An Integrated ontology Development Enviorment for Data Extraction*, Master Thesis, Brigham Young University, 2000.

This thesis describes a portable integrated ontology development environment as a tool to facilitate the creation of application ontologies. The tool provides a method of editing an ORM and its associated data frames and also provides debugging functionality for editing data frames.

[JAVA] The JAVA home page:
http://java.sun.com/

This web site gives information about Java development environment and downloads links.

[KCG+96] L. Karttunen, J-P. Chanod, G. Grefenstette, A. Schiller, Regular Expressions for Language Engineering, *Natural Language Engineering*, Vol. 2, No. 4, pp. 305-328, 1996.

This paper is an introduction to the regular expression calculus, extended with certain operators that have proved very useful in natural language applications ranging from tokenization to light parsing. The examples in the paper illustrate in concrete detail some of these applications.

[Lid95] S.W. Liddle, *Object-Oriented Systems Implementation: A Model-Equivalent Approach*, Ph.D Dissertation, Brigham Young University, 1995.

This dissertation is a collection of papers relating to the topic of object-oriented systems implementation in general, and the development of a new language, Melody, in particular.

[LCC99] S.W. Liddle, D.M. Campbell, and C. Crawford, Automatically Extracting Structure and Data from Business Reports, In

Proceedings of the 8th international conference on Information knowledge management, pp. 86-93, Kansas City, Missouri, November 1999.

This paper presents algorithms that automatically infer the regular structure underlying business reports and automatically generate wrappers to extract relational data.

[Lyo00] R.W. Lyon, *Identification of Temporal Phrases in Natural Language*, Master Thesis, Brigham Young University, 2000.

This thesis presents a method for identifying eight types of temporal phrases in natural language and achieves an overall performance of 86.28%.

[Mik] Mikrokosmos ontology web site: http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html

This web site gives information about Mikrokosmos ontology, in which it defines a knowledge-based machine translation and comprehensive treatment of lexical, ontological, and text meanings in a "society of microtheories" architecture.

[MMM97] A. Mendelzon, G. Mihaila, and T. Milo, Querying the world wide web, *International Jounal on Digital Libraries*, Vol. 1, No. 1, pp. 54-67, April 1997.

This paper presents a query language, WebSQL, that takes advantage of multiple index servers without requiring users to know about them, and that integrates textual retrieval with structure and topology-based queries.

[MS00] A. Maedche and S. Staab, Mining Ontologies from Text, In *Proceedings of 12th International Conference of Knowledge Engineering and Knowledge Management (EKAW 2000)*, pp. 189-202, Juan-les-Pins, France, October 2000,.

This paper presents a general architecture for discovering conceptual structures and engineering ontologies. The authors describe a case study for mining ontologies from text using methods based on dictionaries and natural language text. Their approach combines dictionary-parsing mechanisms for acquiring a domain-specific concept taxonomy with a discovery mechanism for the acquisition of non-taxonomic conceptual relations.

[WN] WordNet home page: http://www.cogsci.princeton.edu/~wn/w3wn.html

This web site gives information about WordNet, including the history of the project, the scope of the database, and the database with source code libraries available for download.

[XmlA] The XML Catalog, listing some academic XML research: http://www.w3.org/XML/

This web site lists various advancements studied on academic fields.

[XmlI] The XML Catalog, listing organizations producing industry-specific XML DTDs: http://xml.org/

This web site lists various companies and organizations that are developing industry specific and cross-industry XML DTDs in various domains.

[YL99] Y. Yang and X. Liu, A re-examination of text categorization methods, In *Proceedings of the 22nd Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 42-49, Berkeley, California, August 1999.

This paper reports a controlled study with statistical significance tests on five text categorization methods: the Support Vector Machines (SVM), a k-Nearest Neighbor (kNN) classifier, a neural network (NNet) approach, the Linear Least-squares Fit (LLSF) mapping and a Naive Bayes (NB) classifier.

IX. Artifacts

This thesis will produce a program that implements the proposed process to generate a data-extraction ontology based on given knowledge bases and Web documents. The program will be written in Java. This research will also produce a general ontology-generating framework, which will include a knowledge base in XML, string recognizers written as regular expressions, and dictionaries. Several generated data-extraction ontologies will also be produced.

X. Signatures

This proposal, by Yihong Ding, is accepted in its present form by the Department of Computer Science of Brigham Young University as satisfying the proposal requirement for the degree of Master of Science.

____
David W. Embley, Committee Chairman

____
Deryle W. Lonsdale, Committee Member

____
Charles D. Knutson, Committee Member

____
David W. Embley, Graduate Coordinator