Assisted Browsing for Semistructured Data

Vineet Sinha
vineet@ai.mit.edu

David R. Karger
karger@theory.lcs.mit.edu

David F. Huynh
dfhuynh@ai.mit.edu

MIT Artificial Intelligence Laboratory / Laboratory for Computer Science
200 Technology Square, Cambridge, MA 02139

ABSTRACT

The development of the RDF[2] standard highlights the fact that a great deal of useful information is in the form of *semistructured data*—objects connected by relations fitting no rigorous schema. To make use of this information it is important to be able to search and browse semistructured data repositories. In this paper, we present a framework, system, and user interface supporting navigation in such repositories. The system is general purpose, involving no hard-coded assumptions about the structure of the repository. The focus is on providing users with helpful "next steps" leading them to the information they are seeking.

Categories and Subject Descriptors

H.3 [**Information Search**]: Information Search and Retrieval; H.3.3 [**Information Search and Retrieval**]: Query formulation; H.4.3 [**Communications Applications**]: Information browsers

Keywords

Navigation, searching/browsing, information retrieval, semistructured data, metadata, semantics

1. INTRODUCTION

The Haystack project[1] seeks to help users visualize and manage their information. To support the customizability and flexibility needed to let users store and navigate through information in the way they want, Haystack provides a semistructured data model in which objects are connected to each other by arbitrary, user-specified relations.

This flexible and generic data model presents several opportunities for improving information retrieval. Searches for information generally involve a dialogue between the user and the computer. The user starts somewhere and follows a sequence of navigation steps (e.g., issuing a query, looking at the results, refining the search parameters, etc.) until the desired information is found. A user interface can help this navigation process by suggesting useful steps to be taken next, such as specifying particular ways to refine the query.

In this paper, we describe a framework for such an assisted navigation system within Haystack. Under this framework, an architect needs not consider the user interface at all and only needs to specify a set of possible navigation steps and their outcomes. The Haystack user interface takes care of presenting these steps to the user and letting him or her select one. Using this framework, we have built what we believe to be a number of *navigation experts*:

select attribute. Given an item, present an *object summary* to allow navigation to other items that share a particular attribute.

expand collection. Given a collection of items, present a *collection summary* to allow an expansion to include other items that are similar to those in the collection.

refine collection. Given a collection of items, narrow down to the subset of items that share a common value on a given attribute.

The experts are intelligent in their own way and represent the various tasks of searching and browsing the user would have to possibly accomplish while trying to reach his or her information goals[5]. Since collections of objects are used pervasively and have more metadata available, navigation experts have also been implemented for collections.

Perhaps the most important aspect of the overall framework is that it is completely agnostic as to the particular metadata in the system; thus, it can continue to work unchanged as users customize their own repositories with new attributes and values they have created or imported from elsewhere. For example, the framework can refine a collection of information based not only on the type of documents or their due dates but also based on project—an attribute that may not have been defined when the system was developed but should be available for navigation once it has been introduced. In contrast, current systems[3] are only designed to work with data whose structure is known at the time of construction and are thus not concerned with issues such as determining which properties associated with the information are important.

2. AN EXAMPLE

Using this navigation framework, Haystack is able to provide a navigation interface for any information provided in RDF[2], the W3C standard for representing semistructured data as triples. As an example of the navigation system, metadata was extracted from the recipes website Epicurious.com, in such a way that a recipe's ingredients are expressed as properties of the recipe, where the property name would be "ingredient" and the property values would be "rice", "egg", etc.

When given a collection of recipes to the navigation framework (Figure 1), the *object summarizer* suggests the user to

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM xxx.

Figure 1: The output provided by the navigation system in Haystack for a collection of recipes.

navigate to a list of objects of the same type, i.e., a collection. The *collection summarizer* notices that all current items have flour and rice as ingredients and therefore suggests a super-collection of items having flour or rice as an ingredient.

With reference to the collection the navigation framework uses the *collection refiner* expert to recommend refining the given recipes by dish types or ingredients. The goal is to find metadata entries in the collection that split the collection along one of many possible orthogonal attributes available. This navigation expert finds all the metadata associated with objects in the collection and then groups the metadata together by the attribute names to suggest possible navigation steps, so that the user is able to get to collections whose members share those metadata.

3. NAVIGATION FRAMEWORK

The navigation framework works by providing an interface to the semistructured data graph as a collection of vectors, such that traditional information seeking techniques from the machine learning and information retrieval[4] communities can be applied in a straight forward manner. Each object gets a vector, with a coordinate for each metadata attribute. Semistructured object attribute values are represented as unique identifiers, and long text strings are split into constituent words and stored with their frequencies.

Annotations on the object properties are used to provide additional information such as the object type, for example, information about whether the properties refer to numeric quantities or dates, so that the user interface can be generated in a more intelligent fashion. Attribute values having ranges are presented using sliders, and the query engine has built-in support for supporting queries with ranges beyond the usual functionality like conjunction, disjunction and negation. Annotations can also be used to tell the navigation framework to have virtual attributes which represent going multiple links away.

Support for collection refinement thus is a direct adaptation of the query refinement technique (cf. [6]).

4. RESULTS

The navigation system was first compared to Epicurious.com, a site run by the publisher of *House and Garden*. The navigation options available on the website are hardwired into its hierarchy such that the navigation steps could not be recommended for an arbitrary collection as obtained when doing operations like advanced search. In contrast, the resulting navigational application, shown in Figure 1, had the advantage of analyzing the current collection information dynamically and was able to present navigation options for any given collection of recipes. The navigation framework did not need to be adapted to the data like those in currently available systems[3].

When the navigation system was tested on the data available in Haystack, i.e., imported e-mail and collections of favorite documents, users were able to refine collections based on the information available. Given a set of objects, document types and user categorization information allowed the refinement expert to suggest refinement by those attributes, as well as more traditional refinement in the case of e-mail where users' were able to refine based on the documents created by a given individual.

The system was also tested on two external datasets: a collection of information about 50 states[1] provided as a comma separated file and an RDF version of the 1999 CIA World Factbook[2]. In both datasets, object properties are encoded as human-readable strings rather than marked up semantically. Thus, we did not expect any interesting results. However, the navigation system recommended navigating to states that have the same birds or flowers, and, from the World Factbook, countries which have the same independence day or currencies. In both of these cases the system would have been able to provide more helpful support for navigation had the data been made available with more semantics. For example, instead of having encoded an area as "114006 sq.mi", it could have been marked up so that its units were in square miles and that the area was "114006". Developing a system that can automatically do these conversions and add structure by learning from the data will make the system more powerful.

5. ACKNOWLEDGEMENTS

Thanks to Kinh Tieu and Dennis Quan for their discussions on this research. This work was supported by the MIT-NTT collaboration, the MIT Oxygen project, a Packard Foundation fellowship and IBM.

6. REFERENCES

[1] Haystack: A personalized information management platform. `http://haystack.lcs.mit.edu/`.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[3] J. English, M. Hearst, R. Sinha, K. Searingen, and K.-P. Yee. Hierarchical faceted metadata in site search interfaces. In *CHI 2002, Minneapolis, Minnesota, USA*, 2002.

[4] D. Harman. Relevance feedback and other query modification techniques. In W. B. Frakes and R. Baeza-Yates, editors, *Information retrieval: data structures and algorithms*, chapter 11, pages 241–263. Prentice-Hall, Inc., 1992.

[5] S. Jul and G. W. Furnas. Navigation in electronic worlds: Workshop report. *SIGCHI Bulletin*, 29(4):44–49, October 1997.

[6] B. Vlez, R. Weiss, M. A. Sheldon, and D. K. Gifford. Fast and effective query refinement. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 6–15. ACM Press, July 1997.

[1]This dataset was extracted from `http://www.50states.com/` and converted by us into RDF.

[2]This dataset is available at `http://www.ontoknowledge.org/oil/case-studies`.