Semantic Navigation Through Semi-structured Information

Vineet Sinha VINEET@AI.MIT.EDU
Dennis Quan DQUAN@MEDIA.MIT.EDU
David F. Huynh DFHUYNH@AI.MIT.EDU
David Karger KARGER@THEORY.MIT.EDU
MIT Artificial Intelligence Laboratory, 200 Technology Square, Cambridge MA, 02139 USA

1. Introduction

The Haystack project seeks to help users effectively 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 semi-structured 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 natural *navigation modes*:

object summary. Given an item, navigate to other items that share a particular attribute.

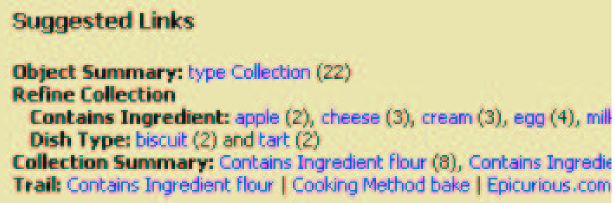
collection summary. Given a collection, expand to include other items that are similar to some of those in the collection.

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

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. This enables the framework to 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 (English et al., 2002) 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 (Quan et al., 2002) is able to provide a navigation interface for any information provided in RDF, a W3C standard for representing semi-structured data as triples. This is a general approach that works for arbitrary data. 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.



Suggested Links

Object Summary: type Collection (22)
Refine Collection
Contains Ingredient: apple (2), cheese (3), cream (3), egg (4), mil
Dish Type: biscuit (2) and tart (2)
Collection Summary: Contains Ingredient flour (8), Contains Ingredie
Trail: Contains Ingredient flour | Cooking Method bake | Epicurious.com

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

When given a collection of recipes (Figure 1), the *object summarizer* suggests the user to 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 subcollection of items having flour or rice as an ingredient.

Since collections of objects are used pervasively and have more metadata available, navigation modes have been implemented for collections. In the figure the navigation framework uses the *collection refiner* navigation mode to recommend refining the given recipes by dish types or the ingredients. The goal is to find metadata entries in the collection that split the collection along one of many possible orthogonal axes available. This navigation mode works by finding all the metadata associated with objects in the collection and then grouping the metadata together to suggest possible navigation steps, so that the user is able to get to collections whose members share those metadata.

3. Navigation Framework

The power of the navigation framework arises from the fact that each navigation mode is intelligent in its own way and that the information made available by the various modes can be put together. The navigation framework, when initialized, uses the properties of the current piece object to set up the various navigation modes. All navigation modes are expected to analyze the current object and produce a list of navigation suggestions for the framework to present. The information produced by the implementation of the navigation modes includes link titles, associated actions resulting from selecting the links, and possibly the number of items that would be found upon selecting each link. The navigation framework groups the possible navigation actions by the navigation modes and presents this information to the user. The framework also allows the object properties to be annotated with additional information, such as whether the properties refer to numeric quantities or dates, so that the user interface can be generated in a more suitable fashion.

4. Results

The navigation system was first compared to Epicurious.com, a site run by the publisher of *House and Garden*. The resulting application, shown in Figure 1, had the advantage of looking at the information dynamically and was able to present navigation options for any given collection of recipes. In contrast, the navigation options available on the website were hard-wired into its hierarchy such that navigation steps could not be recommended for any given collection as obtained when doing operations like advanced search.

When the navigation system was tested on the data available in Haystack, i.e., imported e-mail and collections of favorite documents, the system performed as per expectations. Users were able to refine collections based on either the document types, the categories they had assigned the documents, or in the case of e-mail, 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 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 could 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 by learning from the data will make the system more powerful.

5. Future Work

The navigation system can be improved by using learning to recommend the next steps and the order with which they are presented. Algorithms can analyze the text content in a collection of documents and recommend terms that can be best used for browsing the documents. Similarly, algorithms can also try to learn which items the user will want to click on and then list those items at the beginning of a collection.

Acknowledgements

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

References

English, J., Hearst, M., Sinha, R., Searingen, K., & Yee, K.-P. (2002). Hierarchical faceted metadata in site search interfaces. *CHI 2002, Minneapolis, Minnesota, USA.*

Quan, D., Huynh, D. F., Sinha, V., Zhurakhinskaya, M., & Karger, D. (2002). Basic concepts for managing semistructured information in haystack. *2nd Annual Student Oxygen Workshop, Gloucester, MA, USA.*

[1] This dataset was extracted from http://www.50states.com/ into a comma seperated file and then converted by us into RDF.

[2] This dataset is based on the 1999 World Factbook and is available in RDF at http://www.ontoknowledge.org/oil/case-studies.