View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by UNT Digital Library

End-User Evaluations of Semantic Web Technologies

Rob McCool[1], Andrew J. Cowell[2] and David A. Thurman[3]

[1] Knowledge Systems Lab, Stanford University
robm@ksl.stanford.edu
[2] Rich Interaction Environments, Pacific Northwest National Laboratory
andrew@pnl.gov
[3] National Security Directorate, Battelle Pacific Northwest Division
thurmand@battelle.org

Abstract. Stanford University's Knowledge Systems Laboratory (KSL) is working in partnership with Battelle Memorial Institute and IBM Watson Research Center to develop a suite of technologies for information extraction, knowledge representation & reasoning, and human-information interaction, in unison entitled "Knowledge Associates for Novel Intelligence" (KANI). We have developed an integrated analytic environment composed of a collection of *analyst associates*, software components that aid the user at different stages of the information analysis process. An important part of our participatory design process has been to ensure our technologies and designs are tightly integrate with the needs and requirements of our end users, To this end, we perform a sequence of evaluations towards the end of the development process that ensure the technologies are both functional and usable. This paper reports on that process.

1. Introduction

An often-overlooked element in the software engineering lifecycle is the end user, the individual that shall eventually be using the tool, technique or technology developed for some real purpose. The computer science historical literature is awash with stories of lavishly funded projects that failed to take this stakeholder into account, resulting in systems that fail to meet the exact requirements of their users and are a chore to use. Within KANI, we have used a participatory design process to ensure that our designs and processes are in line with the thoughts of our subject matter experts. Here we report on our iterative evaluation Search on TAP, an application developed at Stanford Knowledge Systems Lab.

2. Search on TAP

Search on TAP is an end-user application that uses documents from the Semantic Web to enhance the search experience beyond the capabilities provided by typical Information Retrieval systems. Search on TAP builds on techniques previously described by Guha *et al.* [2004a] to aggregate information from multiple websites. The information from these websites is translated via scraping from HTML into RDF, and then merged together via a series of owl:sameAs assertions [Guha *et al.*, 2004b]. The end result is a coherent data set that can then be used to perform both entity-based search as well as traditional keyword search. Search on TAP covers 31 source sites over 12 topics, resulting in 188,680 pages containing 1,089,389 entities.

2.1 The User Experience

The use of structured information from the Semantic Web enables users to perform queries that are not possible with a simple keyword based search engine. The types of queries that the Search on Tap engine supports include:

- Entity queries about a single named entity. While keyword search engines can perform such queries, Search on TAP supports disambiguation of named entities. For example, in a query "Harrison Ford", the system can distinguish between Harrison Ford, the modern actor who played Han Solo, and Harrison Ford, the silent film star from the 1920's.

- Attribute queries allow the user to ask for specific attributes of an entity, such as Harrison Ford's birth date, or the population of China. The user may also ask for entities related to another entity, such as querying for "Chicago buildings" or "rail mobile nuclear missiles."

- Comparison queries allow the user to ask for entities that compare in a particular way to another entity. For example, buildings taller than the Sears Tower or roller coasters faster than the American Eagle.

- Group queries support searching for groups of entities, such as countries with a population greater than 250 million or movies starring both Meg Ryan and Tom Hanks.

Search on TAP can be accessed online at http://tap.stanford.edu.

2.2 The User Interface

The user interface for the Search on TAP application is a basic browser-based search interface consisting of a single text entry box labeled "Query" and a

submit button. This is familiar to users and is comparable to other search technologies. As the user types, suggestions are offered to aid the user in automatically completing their query. As shown in Figure 1, this auto-completion feature accelerates the search experience while also educating the user on the coverage of the Search on TAP knowledge bases.[1]

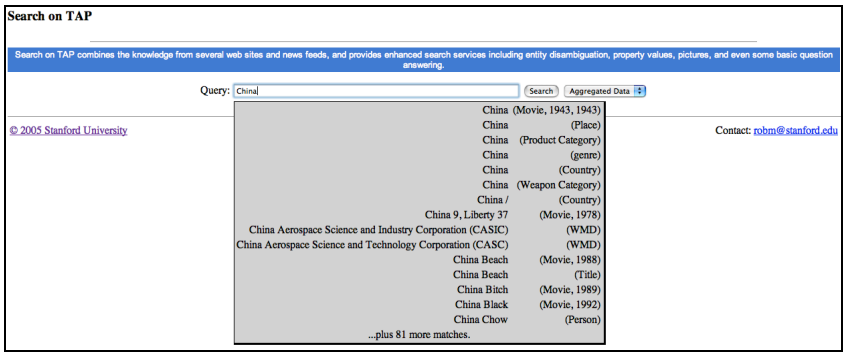

Fig. 1. Entering a query into Search on TAP

This functionality was included as a result of specific user feedback, discussed in section 2.2. Once a user submits a query (by typing a set of words into the query box) the system responds with two columns of results. On the left, Semantic Web-based entities matching the query are displayed, while traditional keyword-search results are displayed on the right (see Figure 2).

Fig. 2. Traditional & Search on TAP results

Displaying results together in this manner enables Search on TAP to add value to the potentially ambiguous keyword results by providing another mechanism to narrow down ones primary search aim. After selecting a specific result, Figure 3 shows the additional query-specific details, including associated sources that are presented to the user.

[1] A full list of sources indexed by Search on TAP is available at http://sp11.stanford.edu/crawl-050210.html

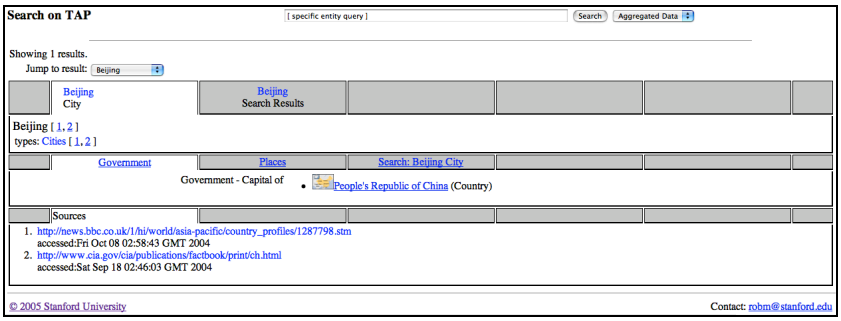

Fig. 3. Detailed information for results

2.3 User Interaction

The Search on TAP user interface is intended to mimic as closely as possible the interface of traditional information retrieval systems such as search engines. Toward that end, the primary query interface is the text entry box, into which the user enters a set of query keywords. This interface approach was taken in lieu of a different approach, such as having the user fill out a form based on the structure of the underlying information, because the structure of the underlying information is very broad. Unlike a structured information source with information about a single domain, such as replacement parts for cellular phones, the Search on TAP dataset is intended to be a prototype for the comprehensive information that will be available on the Semantic Web.

Because natural language processing techniques have proven to be error prone, our approach has been to analyze the query keywords using the structure of the information as a guide. While this does not afford the user the level of flexibility that true natural language processing would provide, we believe it provides a superior user experience than an error-laden natural language processing system. Our philosophy is similar to that which drives Palm Computing to use the "Graffiti" alphabet [Palm, 2005]. Because handwriting recognition at the time was very error prone, Palm developed a pseudo-alphabet that was easier to decode with computing resources at the time, and as a result their product was considered to have a superior user experience compared to the Apple Newton, which tried to perform full handwriting recognition.

In this spirit, Search on TAP analyzes a set of keywords entered by the user using the underlying structured information as a guide. In particular, it looks for words that can represent Classes, Properties, and instances of classes, and tries to find sequences of these representations. For example, an attribute query as described above would look for an instance of a class, followed by a property name that can apply to that class. To ask for Tom Hanks' birthday, for instance, a user can enter "tom hanks birthdate". The underlying Semantic Web information contains an instance of a tap:Person with an rdfs:label of "Tom Hanks", and an rdf:Property

with the rdfs:label "birthdate". The query recognizer can then recognize that the query contains an instance followed by a property, and deduce that the user is asking for the value of the birthdate property for Tom Hanks.

There are 37 such combinations defined in the Search on TAP system, though many of them are redundant. These patterns are used to answer all four types of extended queries described above.

Complications arise in the process of understanding the user's query. The first complication is that one query may have multiple interpretations. In fact, some of them can have thousands of interpretations, particularly when data from movies is included. There are hundreds of movies with very common words as their title, which means many false matches must be processed and removed by the system. Search on TAP uses RDF Schema information to weed out nonsensical pairings. When finding Class-Property-Instance patterns as described above, two types of interpretations are typically removed from consideration.

The first type of interpretation that is removed is the incomplete one. If some of the user's keywords matched structured information, but not all, then that interpretation is removed from consideration. The second type which is removed is one in which the set of classes, properties, and instances do not form anything coherent. This occurs when properties appear which cannot be applied to a class or instance that is also mentioned, or when a class and instance are mentioned and cannot be reasonably connected via property values or chains of property values.

A final complication that arises with this technique is the issue of training. When training a user to use palmOne's Graffiti system, for example, there are 26 characters and several gestures that a user must relearn. Because the scope of the Semantic Web is the full scope of human knowledge, the Search on TAP input system does not have the luxury of such a small set of things to remember. Users must potentially learn hundreds of class names and hundreds of properties. The question of how much of a burden this is to the user is part of what we evaluated in user testing for this tool.

3. Evaluation

A two-step process was used to evaluate the Search on TAP user experience. A heuristic evaluation of the component first allowed us to review the tool against common usability and consistency standards. The elements indicated by the review were re-designed, re-implemented, and reviewed in iterative fashion. After successful heuristic evaluation, the component was evaluated using a group of practicing analysts.

3.1 Heuristic Evaluation

Heuristic evaluation is a usability analysis method that utilizes history and experience to discover problems with a particular component. It involves performing some typical tasks using the component and then noting any disparities between of the component design and a checklist of user interface design principles. The heuristics used in this experiment follow Nielsen's [1994] checklist of design principles. These include ten general principles for user interface design including visibility of system status, consistency and standards, minimalist design, flexibility and efficiency of use, and help and documentation. A study conducted by two usability professionals at Battelle found 17 issues with the initial design. These included issues relating to speaking the users language (e.g., removal of overly technical prose within the user interface), error prevention (e.g., users could potentially enter a refinement query prior to entering the main query) and performance (e.g., a considerable amount of time passed without system status updates).

3.2 User Evaluation

Three Battelle analysts were recruited to help evaluate Search on TAP. They were recruited randomly from an available analyst pool and consisted of two females and one male, all aged within their 30s. The sessions highlighted a number of issues related to customizing such a tool for a specific domain. The participants had difficulty understanding the breadth of information behind Search on TAP. Being able to express that broadness while at the same time maintain expectations (that no tool will be able to answer everything) was a challenge. One solution to this problem was to provide an automatic completion capability (see Figure 1) that guided the user in selecting property names and values. Another highlight from the analyst sessions was the location of the Search on TAP results in relation to the traditional keyword results. Originally, Search on TAP results were shown on the right side of the screen. Due to familiarity with advertisements in popular search engine pages, users indicated they had "learned" to disregard anything on the right-hand side of such a page. The simple solution, shown in Figure 2, was to reverse the columns and provide Search on TAP results on the left side.

4. Summary

The evaluation of a semantic search application using heuristic and user-centered methods provided insight into users' preferences for interacting with semantic information. Subsequent design changes based on user feedback resulted in improvements to the application, leading to a more useful tool for a sample user population. The Search on Tap technology is currently undergoing more formal testing under a government-sponsored evaluation program.

Acknowledgements

The authors would like to thank their respective teams at Stanford University and Battelle Pacific Northwest Division. This work is supported in part by the Advanced Research and Development Activity's Novel Intelligence from Massive Data (NIMD) program.

References

[Guha et al., 2004a] R. Guha, R. McCool, and E. Miller. Semantic Search. In Proceedings of WWW 2004, Budapest, Hungary, 2003.

[Guha et al., 2004b] R.V. Guha, Rob McCool, Richard Fikes, Contexts for the Semantic Web. In Proceedings of the Third International Semantic WebConference, Hiroshima, Japan, 2004.

[Nielsen, J., 1994] Jacob Neilsen. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY.

[Palm, 2005] palmOne, Ways to Enter Data into a palmOne Device. http://www.palm.com/us/products/input/