Open Research Online

Using domain models for context-rich user logging

Conference or Workshop Item

For guidance on citations see FAQs.

oro.open.ac.uk

Using Domain Models for Context-Rich User Logging [*]

Stephen Dignum
School of Computer Science
and Electronic Engineering
University of Essex
Colchester, United Kingdom
sandig@essex.ac.uk

Yunhyong Kim
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
y.kim1@rgu.ac.uk

Udo Kruschwitz
School of Computer Science
and Electronic Engineering
University of Essex
Colchester, United Kingdom
udo@essex.ac.uk

Dawei Song
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
d.song@rgu.ac.uk

Maria Fasli
School of Computer Science
and Electronic Engineering
University of Essex
Colchester, United Kingdom
mfasli@essex.ac.uk

Anne De Roeck
Department of Mathematics
and Computing
Open University
Milton Keynes, United
Kingdom
a.deroeck@open.ac.uk

ABSTRACT
This paper describes the prototype interactive search system being developed within the AutoAdapt project[1]. The AutoAdapt project seeks to enhance the user experience in searching for information and navigating within selected domain collections by providing structured representations of domain knowledge to be directly explored, logged, adapted and updated to reflect user needs. We propose that this structure is a valuable stepping-stone in context-rich logging of user activities within the information seeking environment. Here we describe the primary components that have been implemented and the user interactions that it will support.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Query Formulation; H.5.2 [**User Interfaces**]: Natural Language; I.2.7 [**Natural Language Processing**]: Text Processing

General Terms
Domain Model, Graph Traversal, User Logging

1. INTRODUCTION
Searches within document collections like intranets differ from those within the general World Wide Web [6]. The terminology, structure, and services provided within an intranet are selected to meet organisational requirements, and, consequently, a considerable amount of time is spent by users trying to learn the domain characteristics even before they are able to identify the adequate questions to be submitted to a search system. In the AutoAdapt project, we hope to analyse and accelerate this learning process by implementing a system that presents and logs several domain model representations in response to each stage of a user's logged search activity. By encouraging and logging the direct interaction of users with domain model representations, collective domain user behaviour can be understood within context. The analysed user needs can be incorporated back into the system to adapt domain knowledge representations that are presented to the users, creating a continuous feedback loop.

Provision of domain model knowledge has been shown to aid user search for the information they need [3]. A domain model is effectively a structure that characterises the domain dataset from the domain user perspective, e.g. a graph where nodes represent domain concept terms and edges between nodes their relationship, possibly weighted to express how specific the term is or how closely related the terms are within the collection.

One of the difficulties in using traditional logging of user activity, such as submitted query terms, URL clicks, and page viewing time, to adapt search systems is the lack of sufficient context for identifying the user actions that are truly relevant to the user's information need. We implement methods of explicitly visualising domain models to accompany each search step, in addition to a list of links to search results, and a set of query term suggestions. By concurrently logging user interaction with the these components we have a mechanism to enable subsequent weblog analysis. For example, different document selections following the exploration of the same path may indicate relevance between documents, different paths leading to the same document may indicate relationships between paths, a comparison of path before and after a document selection should yield some understanding of the nature of the document selection.

We present here a working system including a graphical do-

[1]http://AutoAdaptProject.org

main model presentation, a document list and term suggestions designed to capture the described information.

2. RELATED WORK

It is frequently pointed out that users are reluctant to leave any explicit feedback when they search a document collection. However, implicit feedback, e.g. the analysis of log records, has been shown to be good at approximating explicit feedback. For example, users often reformulate their query and such patterns can help in learning an improved ranking function [2]. The same methods have shown to improve an adaptive domain model on a local Web site created using formal concept analysis lattice structures [4].

It has already been evidenced that users want support in selecting search words for query formulation but also it has been recognised that they want to stay in control with respect to making the final decision to submit a query [8]. Furthermore, it has been noted that users like to be provided with system-guided query suggestions even if suggestions are not relevant to the current query [7]. Users have shown signs of being more inclined, in a search environment that supports navigation, to submit new queries, or resubmit modified queries, than to navigate away from the result set [5]. Finally, increased activity in developing interactive features in search systems used across existing popular Web search engines suggests that interactive systems are being recognised as a promising next step in assisting information search. The work proposed in this paper is very much in line with what Belkin calls the *challenge of all challenges* in IR at the moment, to move beyond the limited, inherently non-interactive models of IR to truly interactive systems [1].

3. USER INTERFACE

In figure 1 we can see a screenshot of our demonstration system. There are four basic components, a) a simple entry box for query terms, b) a list of URLs with associated snippets, c) a graph displaying a segment of a domain model, and d) a list of suggested terms for query refinement.

The user enters a set of query terms, this results in a number of documents being returned. Using the query terms, additional terms are automatically extracted, e.g., from the domain model and the highest ranked documents. These terms are then represented as nodes in a graph as a segment of the domain model. The user can then traverse the graph by clicking on the nodes (the effect is to make that term the centre of the graph). On term selection the list of suggested terms is updated to show terms closely related to the selected term. The user can then add the term to the existing query or use it as a new query.

The modular nature of the software allows a standard user interface and logging structure (described in the next section) irrespective of the domain model creation and adaptation algorithms employed. We can, for example, examine different interaction styles and evaluate other domain model visualisation tools.

4. LOGGING INFORMATION

In addition to logging user query terms with presented and selected URLs it was decided to log the segment of the domain model presented to the user. As we intend to modify the domain model over time based on responses to the model presented, it is essential that a complete copy of the presented model segment is retained in the database. Of particular interest is the term positioned at the centre of the graph and the co-ordinates of the other terms. Using this information and the term clicks we can determine how the model was traversed, allowing us to identify which terms were also visible and ignored. Suggested terms (derived by the model) are also recorded along with any selection (to expand, or replace initially submitted query terms).

The logging structure allows us to record a number of user decisions without the need for explicit feedback. For example, the selection of a term in a domain model can provide a ranking of terms, i.e., above those shown but not selected. Also, suggested terms derived from a particular traversal can be ordered. In addition, we can compare sessions that have resulted in the same URL being selected in order to capture related terms or similar portions of the domain model. It is also possible to compare portions of different domain models to discover missed relationships or terms.

5. FUTURE WORK

As the next step, we propose to test the infrastructure in this document across several domain collections and model creation/adaptation algorithms to extensively evaluate the effectiveness of the system in capturing the context of user interaction.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] N. J. Belkin. Some(what) grand challenges for information retrieval. *SIGIR Forum*, 42(1):47–54, 2008.

[2] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *IEEE Computer*, 40(8):34–40, 2007.

[3] U. Kruschwitz and H. Al-Bakour. Users Want More Sophisticated Search Assistants - Results of a Task-Based Evaluation. *JASIST*, 56(13):1377–1393, November 2005.

[4] D. Lungley and U. Kruschwitz. Automatically maintained domain knowledge: Initial findings. In *Proceedings of ECIR*, pages 739–743, 2009.

[5] M. Mat-Hassan and M. Levene. Associating Search and Navigation Behavior Through Log Analysis. *JASIST*, 56(9):913–934, 2005.

[6] M. White. *Making Search Work: Implementing Web, Intranet and Enterprise Search.* Facet Publishing, 2007.

[7] R. W. White, M. Bilenko, and S. Cucerzan. Studying the Use of Popular Destinations to Enhance Web Search Interaction. In *Proceedings of SIGIR'07*, pages 159–166, Amsterdam, 2007.

[8] R. W. White and I. Ruthven. A Study of Interface Support Mechanisms for Interactive Information Retrieval. *JASIST*, 57(7):933–948, 2006.

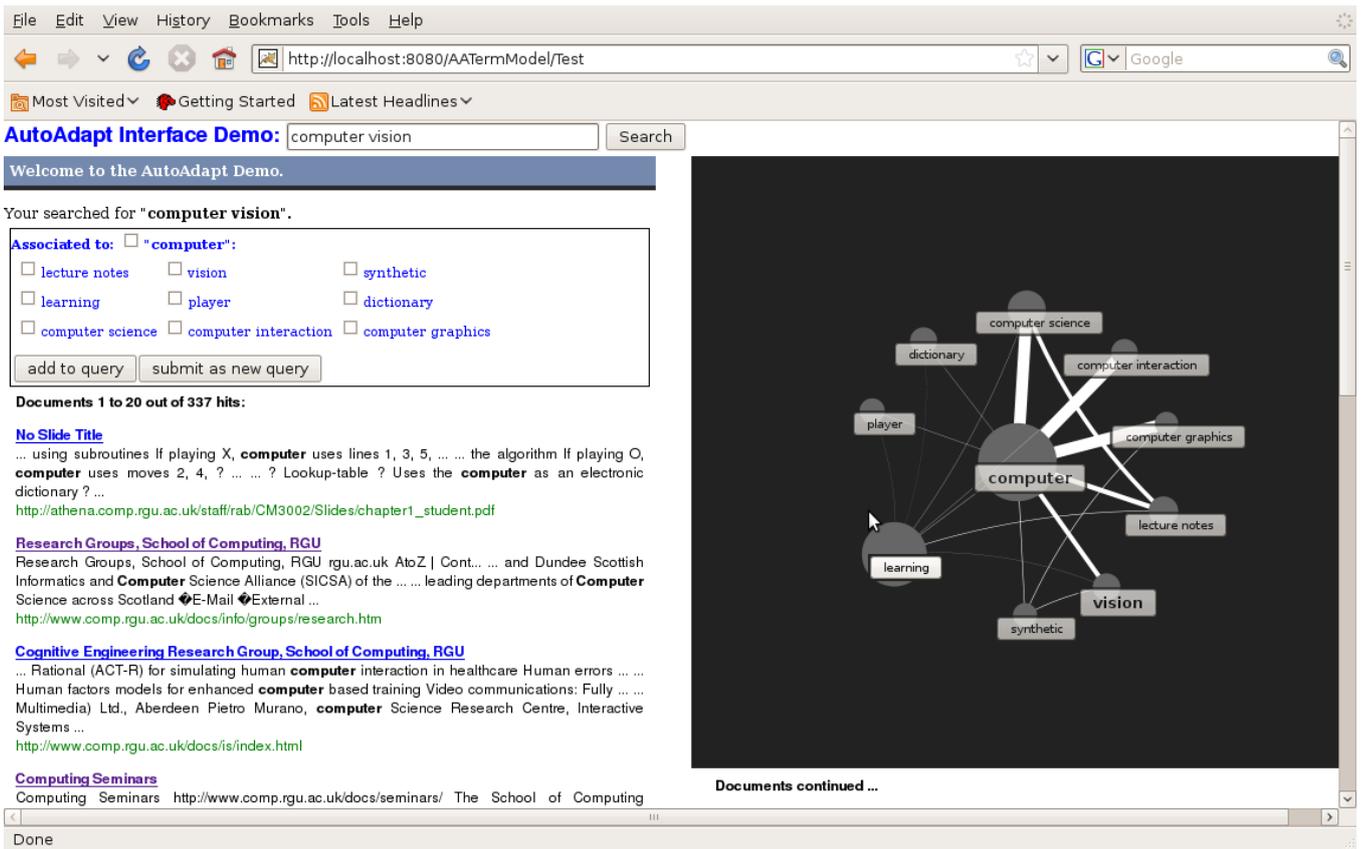[2]http://blog.thejit.org/javascript-information-visualization-toolkit-jit

Figure 1: Screenshot of AutoAdapt Demo System.