Information Seeking Dialogues with Automatically Acquired Domain Knowledge

Udo Kruschwitz
Department of Computer Science, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
udo@essex.ac.uk

1 Overview

Improving Web search technology is a hot topic. One aspect that makes it so interesting is the fact that Web documents are typically not plain text files - instead, they contain a tremendous amount of implicit knowledge stored in the markup of the documents. Much of this need not be used in general Web search, because the search engine doesn't need to understand the documents it is accessing. But what if the document collections are very domain-specific or limited in size? This type of data source is everywhere, from corporate intranets to local Web sites. To help a user search such collections we created a search system based on a generic framework. The system incorporates a simple domain-independent dialogue manager and an automatically created model of the domain.

2 The Domain Model

The simple graph in figure 1 is an actual example from one of our sample domains that gives an idea of how a domain model may be structured. We see a simple tree of related terms that can be used to either assist a user in the search process, perform some automatic query refinements or allow the user to browse the collection. Note, that the *types* of relation in the sample tree are not formally specified. This is significantly different from formal ontologies or linguistic knowledge sources such as *WordNet*.

However, typically a domain model is not available. We construct a domain model automatically by exploiting the markup of documents (see (Kruschwitz, 2001) for detailed information on this process as well as links to related work).

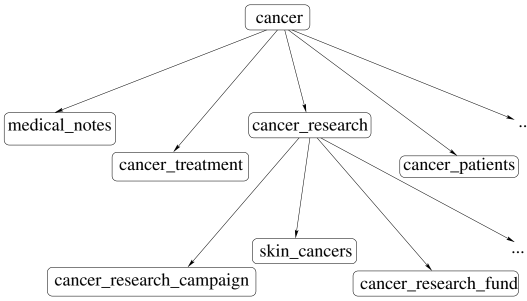

Figure 1: Partial domain model

Having built the model it can now easily be incorporated into our generic framework which is an online search system that has access to a standard search engine and the domain model (Kruschwitz, 2003). This search system is a specialized dialogue system that offers and makes choices about search through the set of documents based on the domain model. This dialogue system is unlike *single shot* systems such as standard search or question answering in that it interacts with the user by offering options to refine or relax the query. With a new document collection coming along we just have to press a button to create the appropriate domain model. The rest of the framework can be left unchanged. Note, that the dialogue manager itself is domain-independent.

3 The Dialogue Manager

We understand dialogue roughly as a movement in the space of document descriptions. That means there are no strictly defined dialogue states as they are commonly used in dialogue systems. Our dialogue states are calculated automatically, and they are represented as a tuple containing a number of different types of information as we shall see further down.

The dialogues we describe are system initiated. Although the user has some freedom to navigate through the dialogue, it is basically an information seeking task that needs to be performed and the system is merely an assistant to help the user get to the right set of answers. As such the focus of attention is a system that presents results alongside possible choices the user might want to consider to continue the search task.

How do we move from one state to the next one? The general idea is to analyze the current dialogue state and the user input in order to generate a new dialogue state. A dialogue state is characterized by: (1) a dialogue history; (2) a formalized user query (called *goal description*); (3) the set of results that matches this goal description; (4) a set of *potential user choices* (options to relax, refine or change the query in other ways).

The list of choices is selected from a set of possible modifications to the goal description taking into account their effect on the result set. This may sound very general. The reason is that we are able to describe a variety of dialogue strategies depending on how those possible choices are to be selected by the system.

Relaxation and refinement steps are the essential building blocks that define the choices a user can select from to continue the dialogue. Ideally, our dialogue system would explore all relaxation and refinement possibilities and select the best ones. However, the reasons for not doing so include the complexity involved and our experience that users are generally not very happy if the system performs complex actions automatically, e.g. automatic query expansion using hypernyms *and* synonyms. Therefore, each of the choices represents a *single* refinement or relaxation step. These choices are derived from the domain model. Customizations of the dialogue manager allow the incorporation of other knowledge sources.

The interaction between user and search system serves the purpose of navigating the user from some initial search request to a satisfactory set of answers. In other words, the *goal description* is constantly being updated in a sequence of simple *dialogue steps* until it matches a set of documents the user is happy with. A dialogue step can be seen as a number of smaller steps to be performed in this order: (1) evaluate the user input; (2) calculate the new dialogue state; (3) perform all actions corresponding to the state transition (e.g. display results and choices).

A *goal description* is a set of attribute-value pairs where each attribute is a document property or a system property and its value is an element from the domain of that property. A goal description is something like a formalized user query. At each stage of a dialogue we can have a look at the goal description and see the current user request. In the simplest case we would just have a list of keywords. In a more advanced search system we have a number of properties that can be tuned and changed during the dialogue.

4 Results and Next Steps

UKSearch is an implemented prototype of our dialogue based search system. Results of a detailed task-based evaluation (using TREC's interactive track guidelines) have shown that users prefer this system over a baseline approach - a standard search system (detailed results to be published shortly). The focus is now on further evaluation steps. An EPSRC grant has been awarded for a 15-month research project entitled *"Investigating the Usefulness of Markup-Based Knowledge Extraction"*[1]. The project has started in June 2003 and will include larger scale evaluations of the search framework for a number of document collections.

References

U. Kruschwitz. 2001. A Rapidly Acquired Domain Model Derived from Markup Structure. In *Proceedings of the ESSLLI'01 Workshop on Semantic Knowledge Acquisition and Categorisation*, Helsinki.

U. Kruschwitz. 2003. An Adaptable Search System for Collections of Partially Structured Documents. *IEEE Intelligent Systems*, July/August:44–52.

[1] EPSRC grant number GR/R92813/01