Squirrel: The SEKT project Search and Browse Tool

Alistair Duke
BT, Next Generation Web Research
Adastral Park, Martlesham Heath, Ipswich IP5 3RE, United Kingdom
alistair.duke@bt.com

Abstract

Squirrel is the Semantic Search and Browse tool developed in the EU SEKT[1] (Semantically Enabled Knowledge Technologies) project. The key principles of Squirrel are firstly a departure from presenting the results of a query as a ranked list of documents towards one where knowledge contained within documents is presented to the user, secondly the user is able to search and receive results based upon an ontology which is populated by SEKT ontology and metadata generation tools and thirdly that the users context is taken into account when a search is carried out.

The demonstration shows how various search and browsing strategies are handled by Squirrel and how discovered knowledge is then presented to the user.

1 Introduction

The generally accepted view is that search engines based on conventional IR techniques (employing keyword and phrase matching between the query and index) alone tend to offer high recall and low precision. The user is faced with too many results and many results that are irrelevant. The main reason for this is the failure to handle polysemy and synonymy.

Although it is possible to disambiguate queries by adding more terms and by making use of more sophisticated functions (e.g. Boolean operators) there is evidence to suggest that the majority of users do not do this. Other techniques to help the user disambiguate their query are not generally offered by conventional search engines, neither is any attempt to determine the context of the searcher. These search engines generally treat search in isolation: the results from a search engine for a given query are identical, being independent of the user submitting the query and the context in which the user made the request.

The context of the person making a search can help to determine the relevancy of documents in the result set. Some commercial search engines offer the user a choice from a set of subjects determined from the classification of the result set. The classification of each page is generally determined via a manual process; the manual process can take time and is obviously subjective. There are also efforts to determine the context of the user dynamically. The identity of the individual, the place, the time and the purpose are all factors contributing to the overall context. These dynamic approaches (described in [Thurlow et al., 2004]) have been found to improve precision. A key aspect of the SEKT approach to knowledge access is to exploitation of context.

The addition of semantic information is expected to improve search and browse services. In its most simple form it will enable a user to search and browse topics classed and linked in accordance with a domain-specific ontology. Simple semantic web-based search engines are expected to improve the precision of results through attribute based searching of a repository of semantic annotations [Thurlow et al., 2004] [Davies et al., 2002].

The Squirrel tool is a Search and Browse tool developed in the EU SEKT project. The key principles of Squirrel are firstly a departure from presenting the results of a query as a ranked list of documents towards one where knowledge contained within documents is presented to the user, secondly the user is able to search and receive results based upon an ontology which is populated by SEKT ontology and metadata generation tools and thirdly that the users context is taken into account when a search is carried out.

The demonstration shows the application of Squirrel to the Digital Library case study in SEKT.

4. Implementation of Squirrel

Squirrel can be defined as a knowledge access tool in that it provides the user with a means to gain access to the contents of a knowledge base (which they may or may not be familiar with). In the SEKT project, knowledge access tools builds upon components that provide automated ontology and metadata generation, reasoning and user profile construction. Squirrel exists as a search and browse interface and accompanying server. The server receives queries from the interface and forwards these to an ontology store and reasoning engine (KAON2)[Motik et al., 2004]. Before forwarding the queries, however, the server first obtains the user's profile from the store in order to take into account the short and long interests of the user. This information is used to expand or constrain queries as appropriate and also to modify the presentation of results to the user.

[1] http://www.sekt-project.com

The search interface allows the user to follow a number of search strategies. At the highest level, these can be classified as search or browse. Search in general allows the user to enter a query based upon their information need. Browse in general allows users to look around the available knowledge when they do not have a clear idea of what they are looking for.

2.1 Search

Considering search first, the interface first allows users to enter queries in the form which is utilized by traditional web search (and which most have become used to) i.e. a text box. The terms entered by the user are first checked against items in the ontology. In the application of Squirrel to particular domains it is expected that knowledge about specific entities will be best presented in a certain way e.g. in the Digital Library context, the search might match against the name of a known author. The presentation can be predetermined by a knowledge engineer, through an administration interface, who can create a template based upon the ontological properties of the entity; so the presentation for an author might show the authors details but also extracts from the most recent publication of the author (and it might also hide information deemed not appropriate for the user). This presentation can go further with the application of rules to the results e.g. you might to show which authors have collaborated with the found author. A simple rule can be specified to find authors who have co-written papers which can then be evaluated by the reasoning engine.

In the case where a match occurs against two very different entities, Squirrel uses profile information such as context and preferences to determine the most likely match but also considers the frequency of occurrence of the entity, etc. Crucially, the user is also offered the chance to change the decision made by the system.

Often matching entities will not have predetermined display templates, particularly in the case where ontologies and metadata are automatically learned from the corpus. In this case Squirrel presents the user with to opportunity to constrain their search based upon the nature of the learned ontology.

The user is further supported by clustering query results and allowing the user to constrain their search based upon parts of the cluster. Here, again, Squirrel builds upon a SEKT component – in this case TextGarden[2].

Finally, a consolidation of the results is presented to the user. This includes appropriate facts extracted from the knowledge base that are relevant to the search and key features from the text that contain relevant entities or have a close relationship with the user's profile.

2.2 Browse

Users who browse generally do not have a clear idea of what they are looking for and require the interface to help them understand what is available. In the digital library example, users might find it useful to browse around the various topics that items are classified under or to choose what kind of resource they are looking for e.g. a journal paper or and author. In the case of topics, a hierarchy can be presented to the user but this can become unwieldy for large topic sets and where it is necessary to present relationships outside of those that can be shown in a traditional hierarchy.

In Squirrel, a topic hierarchy is available but there is also a topic chooser which allows the user to specify imprecise terms which are matched against topics but then allow them to browse around related topics (where relationships have been inferred with manually or automatically).

Users are able to add and remove constraints to their browsing in order to modify the depth and breadth of the results matching their browsing query.

3 Conclusion

This demonstration of Squirrel shows a search and browse tool developed in the SEKT project. The tool comprises of a search interface and a server which makes use of other SEKT project components such as a user profiler and reasoning engine. An administration interface allows the presentation style of results to be pre-determined. The demonstration used data from the SEKT Digital Library case study.

References

[Thurlow et al., 2004] Ian Thurlow, Nick Kings, and John Davies. "*Knowledge Access on the Semantic Web"* SEKT Project, 2004, Deliverable Reference D.5.1.1. http://www.sett-project.com

[Roman et al., 2005] Dumitru Roman, Holger Lausen, Uwe Keller. *"D2v1.2 Web Service Modelling Ontology (WSMO)"*, WSMO Working Draft 13 April 2005. http://www.wsmo.org/TR/d2/v1.2/

[Davies et al., 2002] John Davies, Uwe Krohn and Richard Weeks. *"QuizRDF: search technology for the semantic web"* WWW2002 workshop on RDF & Semantic Web Applications, 11th International WWW Conference WWW2002, Hawaii, USA, 2002

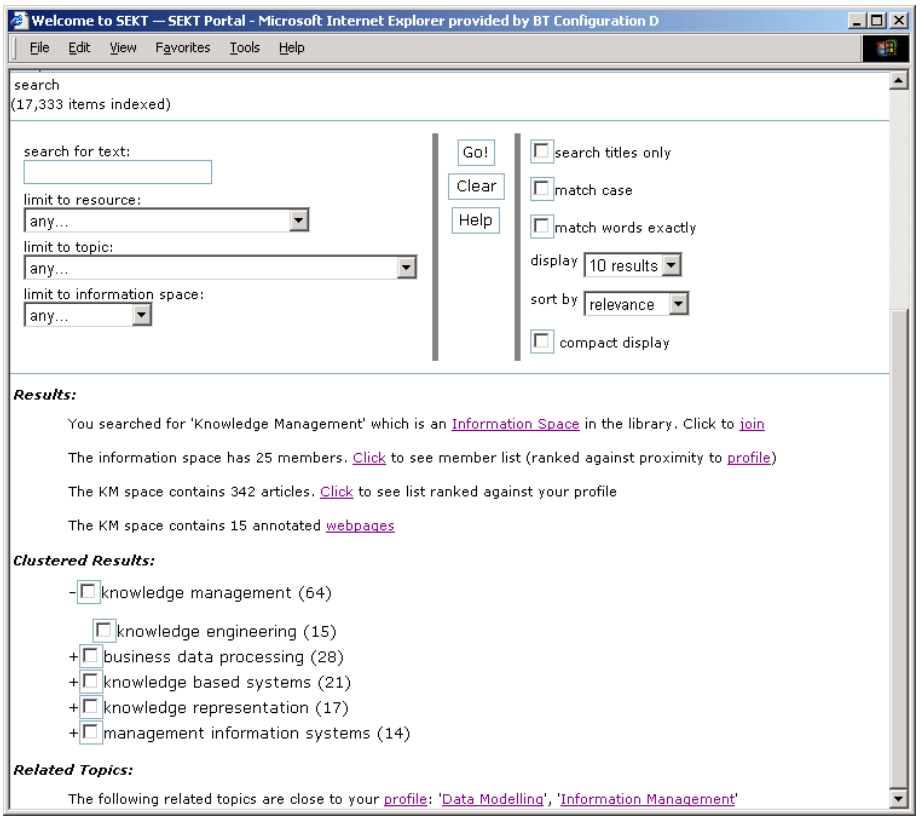
[Motik et al., 2004] Boris Motik, Ulrika Sattler and Rudi Studer. ***"****Query Answering for OWL-DL with Rules"****.*** *Proc. of the 3rd International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, November, 2004, pp. 549-563

[2] http://ist-world.dfki.de/Links/textgarden

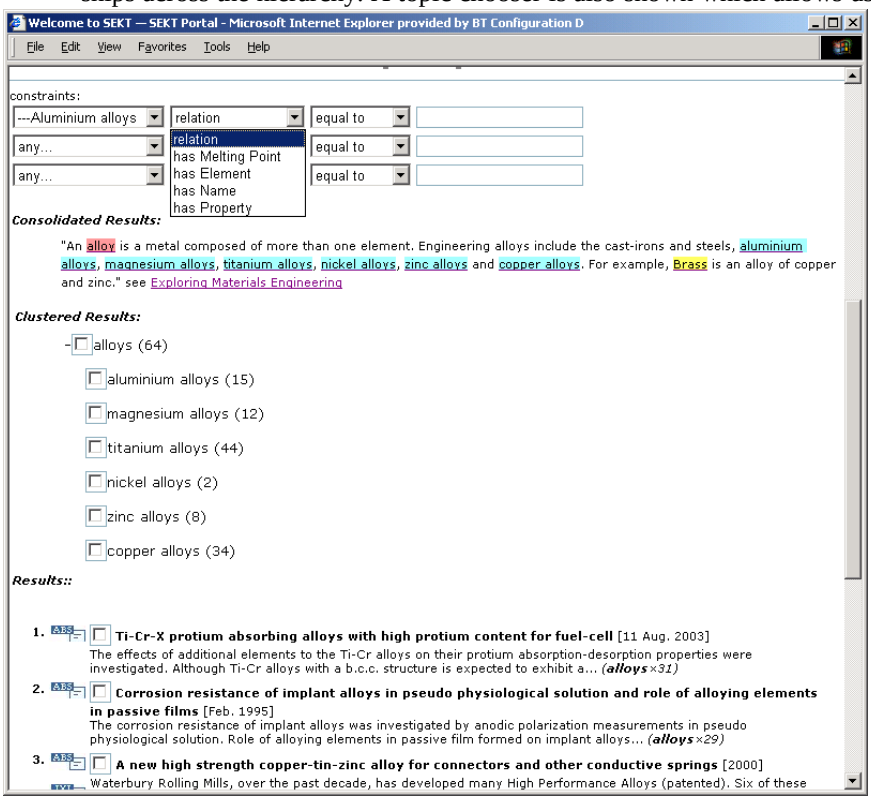
Demonstration Description

The demonstration will show the application of the search tool to Digital Library data. The demonstration involves three main elements

1. A search session. Here the demonstration shows the user entering terms in the search interface. Key aspects are the identification of known entities in the query and the appropriate presentation of results based upon this (as shown in the first screenshot). Query disambiguation is also introduced here as are clustering of results, consolidation of results and extracted knowledge. Name entities are highlighted in the results. Users are also able to constrain their search based upon extracted ontologies and metadata (as shown in the second screenshot).



2. A browse session. Here the user is able to browse through topics in the library arranged in a hierarchical fashion but with relationships across the hierarchy. A topic chooser is also shown which allows users to find the appropriate part of the hierarchy more quickly



3. An admin session. The admin tool allows the creation of templates which determine how particular entities are displayed to the end user. The knowledge engineer is able to re-order the displayed fields
   - omit displayed fields
   - re-name the properties
   - group similar fields
   - make literal fields searchable
   - combine the instances of several entities into one results e.g. author plus publications and collaborators.