Survey of Semantic Search Research

Eetu Mäkelä

Semantic Computing Research Group,
Helsinki Institute for Information Technology (HIIT),
Helsinki University of Technology, Media Technology, and University of Helsinki
eetu.makela@tkk.fi, http://www.cs.helsinki.fi/group/seco/

Abstract. This paper surveys the research field of semantic search, i.e. search utilizing semantic techniques or search of formally annotated semantic content. The survey identifies and discusses various prevalent research directions in semantic search, as well as extracts common methodology used in them.

1 Introduction

This paper surveys the research field of semantic search, defined for the purpose of this paper as either search utilizing semantic techniques or search of formally annotated semantic content. The survey is based on reading and exploring some 20 different papers and approaches to semantic search. The material was gathered based on a keyword search for 'semantic AND search' in various publication databases, as well as by going through references in papers already found, and is meant to be a sufficient representative of the variety of semantic search related research.

From the data gathered in the survey, a number of prevalent research directions in semantic search were identified, based on similarity of research goals. These, as well as the individual approaches that are part of them, are described in chapter 2 of this paper. Besides research directions, the papers were also analyzed for common methodology. The methods used in the various approaches are noted when they are discussed, but the actual descriptions of the common design patterns are located in chapter 3. The paper ends with a chapter drawing conclusions.

2 Research Directions in Semantic Search

From the corpus of research utilized in this survey, five distinct research directions emerged. While the categories sometimes do not differ much in methodology, they seem sufficiently separate and coherent with regards to research goals, to be an accurate clustering of the research space. The five directions presented are: augmenting traditional keyword search with semantic techniques, basic concept location, complex constraint queries, problem solving and connecting path discovery. All of these are described in detail in the following subchapters.

2.1 Augmenting Traditional Keyword Search with Semantic Techniques

Much, particularly early research on semantic web enabled search deals with augmenting traditional text search with semantic techniques. This research direction differs significantly from the others presented later in the sense that it does not usually presume the bulk of the actual knowledge being sought to be formally annotated. Instead, ontological techniques are used in a multitude of ways to augment keyword search, whether to increase recall or precision. In the following, a variety of approaches is presented:

Many query expansion implementations utilized in keyword search make use of thesaurus ontology navigation as a step in query expansion. Particularly utilized is the large WordNet ontology, defining synonym and meronym sets for words. These systems all function along the same basic scheme: first, the keywords are located in the ontology, then, various other concepts are located through graph traversal, after which the terms related to those concepts are utilized to either broaden or constrain the search. In [1] and [2], terms are expanded to their synonym and meronym sets using the boolean OR operations available in most search engines. In Clever Search[3], a particular meaning of a word in the WordNet ontology can be selected, resulting in the clarification text of that meaning being added to the search keywords via the boolean AND operator. In the ontology navigation phase, the implementations differ mostly in which properties of the ontology are navigated and which terms are picked.

A very simple manner of augmenting traditional keyword search results is taken in the “Semantic Search” interface[4] of the TAP infrastructure. Here, in addition to a traditional keyword search targeted at a document database, the keywords are matched against concept labels in an RDF repository. Matching concepts are then returned in addition to the located documents. The paper also posits a continuation of the search similar to the one described in [3], where, if multiple concepts match the keyword, the user can select his intented meaning to constrain the search. Here, however, the idea is not to expand search terms, but to use some procedure to classify the actual documents as pertaining or not pertaining to concepts, and constraining results based on that semantic annotation.

[5] describes an algorithm for locating additional information relevant to a query given a starting set obtained via text search. First, traditional text search is applied into a document collection. Then, a process of RDF graph traversal is begun from the annotations of those documents. The intention is to find related concepts such as the writer of the document, the project the document refers to, etc. in a general manner. The traversal is done by a spread activation algorithm, for the use of which the arcs in the ontology are weighed according to general interestingness. This is calculated by combining a specificity measure favoring unique connections in the knowledge base, and a cluster measure, which favors links between similar concepts.

The CIRI[6] search system provides an ontological front-end to text search. The search is done through an ontology browser that visualizes the ontologies created for search as subsumption trees, from which concepts can be selected to constrain the search. The actual search is done through keywords annotated to these concepts, as well as subconcepts, utilizing a traditional text search engine and boolean logic. The actual search algorithm is in many ways similar to the query expansion algorithms discussed before. The main difference is in the user interface being based on direct ontological browsing, leaving out the first step of mapping a search keyword to the ontology.

2.2 Basic Concept Location

While much of semantic search research is directed at adding semantic annotations to data in order to improve search precision and recall on that data, there are other reasons for writing down information with formal semantics. Therefore, some research begins with the assumption of concepts, instances and relationships, and deals with the task of efficiently locating instances of these core semantic web datatypes.

Usually, data on the semantic web is divided into two classes: ontological and instance data. The actual data the user is interested in are individuals belonging to a class, but the domain knowledge and relationships is described primarilty as class relationships in the ontology. This organization of data points to a natural way of locating information, exemplified for example in the SHOE search system[7]. In SHOE, the user is first given a visualization of the subsumption tree of classes in the ontology, from which he can choose the class of instances he is looking for. Then, the possible relationships or properties associated with the class are sought, and a form is presented that allows the user to constrain the set of instances by applying keyword filters to the various instance properties. When the properties point to objects, the target of the filtering will be the label of the referenced resource. Queries that can be formulated via this paradigm are e.g. “find all publications with a particular author name, from a particular project”. A similar approach is also taken in some versions of the SEAL portal tool[8].

The class subsumption-tree -based approach here is similar to the single facet search used in many Internet directories such as dmoz.org and Yahoo!. A more powerful paradigm is that of multi-facet search[9]. This is the approach behind the main search of the OntoViews[10]-based portals, and the SWED[11] directory portal. In multi facet search, multiple distinct views are provided into the data. For example, in the OntoViews -based MuseumFinland portal[12], where the information items are museum objects, the user is given views such as Object Material, Place of Manufacture and Context of Use. These views are created via ontology projection, utilizing also the various other hierarchical relationship trees and leaf relations usually inherent in ontologies besides class subsumption and membership.

Here, the idea is that the user can start constraining their search from the view that is most natural to them. Additionally, constraints on the different views can be combined to create more complex queries, so the user can for example search for museum objects manufactured in China and used in Fishing. Additional implementations of the idea include the Longwell browser of the Simile project[1], which differs in that it is restricted to flat views.

In some versions of the OntoViews semantic portal creation tool[10], a concept called semantic autocompletion[13] is utilized, which makes use of keyword search as a prelude to ontological navigation. The idea is taken to its furthest in the Veturi yellow pages service discovery portal[14], where the main interface of the portal opens with a keyword field. The keywords, however, are not linked directly to information items, but

[1] http://simile.mit.edu/longwell/

to ontological classes in the different views, from which semantic disambiguations can be made. The search then proceeds as a multi-facet search query.

Once the search has proceeded to the point where at least a single interesting instance is located, additional information can be retrieved via browsing. The process is analogous to the browsing of web pages linked together via hyperlinks. However, here the items shown are resources and the links between them are defined by their relations. In the simplest case, one concept is shown at a time, along with its properties derived straight from the RDF triples. If a property points to another resource and not a literal, then clicking on that property will browse to the referenced concept.

The authors of the Haystack information management tool[15, 16] base their user interface paradigm almost completely on browsing from resource to resource. This is argued by search behavior research[17], in which the authors posit that actually most searching is done via a process they call orienteering. Here, the premise is that searchers usually don't actually themselves know or remember the specific qualities of what they are looking for, but have some idea of other things related to the sought item. The process of search is then a browsing experience in which the searcher looks for information resources that he knows are somehow related to the target, and from there locates additional information on the target resource until it can be located. An example in the article is of a person searching for a particular piece of documentation. Not remembering where it is stored, she only remembers that it was referenced to in some email message from a co-worker. She then scans through her mails in her inbox, remembering the co-worker who the mail was probably from, locates the correct message and from there extracts the location of the documentation. To facilitate locating points of entry for orienteering, Haystack provides a simple text search interface, based on the rationale that the things people remember about resources are probably their labels or phrases contained in them.

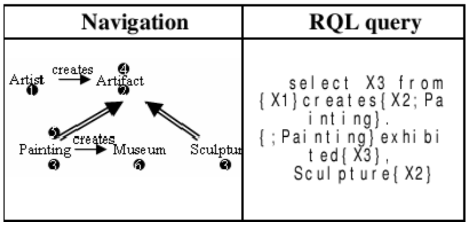
The OntoViews-based portals also offer a browsing functionality between individual information items. This is realized through formalizing interesting RDF path patterns as Prolog rules, and linking to items at the endpoints of pattern-fitting paths beginning at the current item. This allows for linking complexly related items to each other, such as linking a person to all his or her distant relatives.

2.3 Complex Constraint Queries

Many kinds of complex queries can be formulated as locating a group of objects of certain types connected by certain relationships. In the semantic web, this translates to graph patterns with constrained object node and property arc types. An example would be “Locate all toys manufactured in Europe in the 19th century, used by someone born in the 20th century”, where “toys”, “Europe”, “the 18th century”, “someone” and “the 19th century” are ontological class restrictions on nodes and “manufactured in”, “used by” and “time of birth” are the required connecting arcs in the pattern.

While such patterns are easy to formalize and query in the context of the semantic web, they remain problematic because they are not easy for users to formulate directly. Therefore, much of the research into complex queries has been on the level of user interfaces for creating such query patterns as intuitively as possible.

[18] presents GRQL, a graphical user interface for building graph pattern queries that is based on navigating the ontology. First, a class in the ontology is selected as a starting point. All properties defined as applicable to the class in the ontology are then given for expansion. Clicking on a property expands the graph pattern to contain that property, and moves selection to the range class defined for that property, e.g. clicking the creates property in an Artist class creates the pattern "Artist → creates → Artifact", and moves focus to the Artifact class, showing the properties for that class for further path expansion. In addition to lenghtening the path, other operations can be performed on the query pattern. The pattern can be tightened to concern only some subclasses of a class, as in tightening the previous example to "Artist → creates → Painting or Sculpture". In a similar way, property restriction definitions can be tightened into subproperties. More complex queries can be formulated by visiting a node created earlier and branching the expression there, creating patterns such as the one visually depicted in figure 1 which could be used to find all artists that have either created paintings good enough to be exhibited at a museum, or any sculptures, as well as those sculptures, paintings and museums.

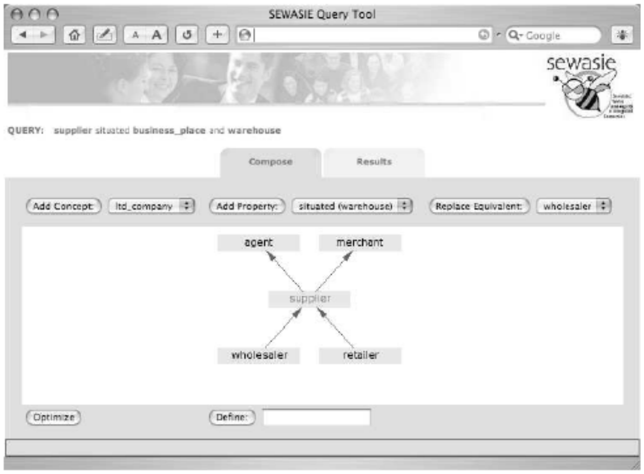


```
select X3 from
{X1}creates{X2;Pa
inting}.
{;Painting}exhibi
ted{X3},
Sculpture{X2}
```

Fig. 1. A visual formulation of a query in the GRQL interface, along with the generated query language expression

Another graphical query generation interface is described in [19]. Here, the user is given some pre-prepared domain-specific patterns to choose from as a starting point, which he can then extend and customize. The refinements to the query can be either additional property constraints to the classes, e.g. "Industry with sector Agriculture" or a replacement of a class in the pattern with another compatible one, such as a sub- or superclass. This is done through a clickable graphic visualization of the ontology neighbourhood of the currently selected class, as shown in figure 2.

The Multi-Facet search portals mentioned earlier can also be thought of as user interfaces for creating a very constrained subset of complex graph patterns. While in the simple case the query is formulated as searching for an information with particular properties (place of use, material, etc.), in a wider sense the definitions of how the objects map to the views can be arbitrarily complex and involve graph navigation, as for example where museum items are not directly annotated to particular event types, but the link is drawn from a combination of item type and material, for example. The portals based on OntoViews also provide limited support for a statistical view to the



QUERY: supplier situated business_place and warehouse

Add Concept: ltd_company

Add Property: situated (warehouse)

Replace Equivalent: wholesaler

Fig. 2. The SEWASIE visual tool for query formulation support

data, because they can group the result set according to a selected category tree or other grouping definition. This provides the portals with the capability to answer questions such as: “From which parts of the world do toys used in 18th century Finland come from”.

In complex queries where the selection is based on a global intersection of distinct selectors, the individual constraints need not be ontological. In OntoViews -based portals, categories and items can be filtered using keyword constraints, while [20] contains a method that allows one to treat keyword search terms as ontological classes whose instances have fuzzy membership values. A fuzzy logic formalism is then used to calculate relevance with respect to the entire query pattern formalized as a fuzzy logic statement.

2.4 Problem Solving

Describing a problem and searching for a solution by inferring one based on ontological knowledge is one of the core use cases often associated with the vision of the semantic web. However, actual implementations of such are rare and they are usually quite simple.

[21] describes a query language for the semantic web, which, despite mostly being intended for simpler SQL-like queries is based on a DL-reasoner, and allows for a form of “if-then” queries. This functionality in turn has been used in the Wine Agent demonstration portal[2]. Here, the user enters information on the flavors in a dish, and the system infers from the ontological knowledge a recommendation for a wine suitable to complement those flavors.

[2] http://onto.stanford.edu:8080/wino/index.jsp

2.5 Connecting Path Discovery

While usually property relations are used to traverse from an interesting resource to another, sometimes what is interesting are actually the paths in the graph connecting the items. In the realized vision for the semantic web, a huge amount of varied semantic data will be available to be mined for semantic connections. An example of a domain where this could prove very useful is the national security domain, where there is a need for locating connections and patterns suggesting possible security threats, such as emerging links between known terrorists and potential recruits.

A major problem here is how to define a link interestingness measure in a way which both eliminates uninteresting relations ("Company A and terrorist organization B are related because they both operate in the same country") but is still general enough to be of use in locating complex, hidden relationships in the data. One attempt at formulating an easily calculable general purpose requirement for interesting associations is described in [22].

3 Common Methodology

While surveying the field of semantic search research, some common methodologies appear. Some are inherent to the RDF formalism and will probably be present in all semantic web -based applications while others are more tied to the search domain. It is posited that knowledge and understanding of these common methods as well as how they are used in the various actual approaches will be of use in planning future approaches to the dilemmas of semantic search.

3.1 RDF Path Traversal

Because the data model of RDF forms a net, where arcs and paths encode information, it is only very natural to apply various forms of net traversal in semantic search.

There are a few primary uses of network traversal found in this survey. One is the location of additional relevant information instances given a starting instance in the net, as in [5]. Another use is in query formulation, such as in the GRQL[18] and SEWASIE[19] query formulation interfaces, where navigation happens at the level of the ontological information about the domain, constraining the query by selecting classes and relationships to be used as constraints in the actual search for instances.

Simple path traversal is also usually utilized when gathering all the information pertaining to an item for visualization. This is again due to the way the RDF data model works, with information important to the user also found in other resources linked to an information item, and not just the direct properties of that item. At least OntoViews, SEAL[8] and Semantic Search[4] all make use of graph patterns for gathering the information to be shown for an item.

3.2 Keyword ↔ Concept Mapping

Mapping between keywords and formal concepts is a common pattern appearing in semantic search. There are a number of reasons for its prevalence. The first is that a presupposition of all knowledge sought being formally encoded is blindly optimistic. Much research is specifically about how to combine searching through textual material with search through formally defined information.

A second reason is that natural language is the form of expression that comes most naturally to humans. Mapping patterns in the graph to sentences, such as in [19] can give the user a clearer picture of what the relationships represent, and the other way round, the user may be more comfortable in formulating his queries as natural language sentences, made use of for example in the Veturi portal[14]. Here, keywords provide an entrypoint for locating information quickly. Keywords and other textual/numeric restrictions can be quickly typed into search fields, contrasted with graphically navigating the ontology in order to locate the concepts and graph patterns to be used as search constraints.

3.3 Graph Patterns

Whether described via RDF path languages or in logical languages, graph patterns are an important concept in semantic search, used in multiple different functions. First, because of the way the RDF data model is organized, graph patterns are often used to formulate and encode such complex constraint queries as discussed in chapter 2.3, specifying and locating interesting subgraphs in the RDF network.

In some systems, such as OntoViews[10], general RDF path patterns are also used to link interesting resources to each other, or, as in [22], to formulate patterns for locating interesting connecting paths between named resources. Also, in result visualization, the parameters on where to fetch information pertaining to the item is also usually given as simple graph patterns.

3.4 Logics

Logics and inference are integrally tied to the larger vision of the semantic web. However, while OWL[3], the newest ontology language standard is for example on large part based on Description Logics, very few actual applications currently make full use of inference based on it or actually any other rule system, and many of those that do, such as [21], could have also been developed using simpler graph patterns.

Mostly this is due to inference on the semantic web being a hard problem. The fact that the the semantic web is designed to work under an open world assumption, while most well explored logics operate well only on a closed world. Also, the vision of the semantic web supposes a large amount of data, a problem for most current inference algorithms.

3.5 Fuzzy Concepts, Fuzzy Relations, Fuzzy Logics

In the research direction of augmenting text search with ontology techniques, there is a need for formalisms which allow for combining fuzzy annotations based on text search with the firmness of semantic annotations. As a result, a number of formalizations for,

[3] http://www.w3.org/2004/OWL/

and experimentions with fuzzy logics, fuzzy relations and fuzzy concepts have been undertaken in that field. [20] is an example.

Fuzzy logics are, however, not only useful in combining text search with ontologies. On the search method research side not directly tied to actual applications, [23] applies fuzzy qualifiers to complex constraint queries, while in [24], the idea is presented that user profiling could be used as a basis for weighting the interestingness of an ontological relation to be used in the search. In [25], a basis is depicted for calculating overlap values for historical and current geographic places, for use in the fuzzy mapping of the concepts to each other in any ontological search.

4 Discussion

There are many common patterns found in the approaches described in this survey. On the technique level, it would seem that in the context of working within an RDF model, quite many of the common methodologies utilized are general, separatable modular steps, which could quite feasibly be made use of in most of the systems, regardless of research direction.

But not only are the methodologies general, it would seem that some of the research directions can be combined. Simple concept location can be seen as a precursor and subset of the interfaces allowing selection by more complex graph patterns. Fuzzy logic formalisms and fuzzy concepts allow for the combining keyword search results as equal partners in complex constraint querying. And while usually complex constraint queries have focused on models where individuals and classes are the interesting information items, also the relations are present as equal partners in all the graph pattern, path and logic formalisms. And after finding a result set using complex constraints, there is no reason not to apply the graph traversal algorithms to locate additional result items.

The only direction that does not neatly wrap into the others in this way is inference-based problem solving. While it can always be said that any search problem is indeed a problem to be solved, the leap here would be much longer. While for example the OWL-QL query language[21] is based on reasoning, reasoners and inference seem in general a much heavier tool than need be for the most usual cases of semantic search.

References

1. Moldovan, D.I., Mihalcea, R.: Using wordnet and lexical operators to improve internet searches. IEEE Internet Computing **4** (2000) 34–43
2. Buscaldi, D., Rosso, P., Arnal, E.S.: A wordnet-based query expansion method for geographical information retrieval. In: Working Notes for the CLEF Workshop. (2005)
3. Kruse, P.M., Naujoks, A., Roesner, D., Kunze, M.: Clever search: A wordnet based wrapper for internet search engines. In: Proceedings of the 2nd GermaNet Workshop. (2005)
4. Guha, R., McCool, R., Miller, E.: Semantic search. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, ACM Press (2003) 700–709
5. Rocha, C., Schwabe, D., de Aragão, M.P.: A hybrid approach for searching in the semantic web. In: Proceedings of the 13th international conference on World Wide Web. (2004) 374–383

6. Airio, E., Järvelin, K., Saatsi, P., Kekäläinen, J., Suomela, S.: Ciri - an ontology-based query interface for text retrieval. In Hyvönen, E., Kauppinen, T., Salminen, M., Viljanen, K., Ala-Siuru, P., eds.: Web Intelligence: Proceedings of the 11th Finnish Artificial Intelligence Conference. (2004)
7. Heflin, J., Hendler, J.: Searching the web with shoe (2000)
8. Maedche, A., Staab, S., Stojanovic, N., Studer, R., Sure, Y.: Seal - a framework for developing semantic web portals. In: Advances in Databases, Proceedings of the 18th British National Conference on Databases. (2001) 1–22
9. Mäkelä, E., Hyvönen, E., Sidoroff, T.: View-based user interfaces for information retrieval on the semantic web. In: Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction. (2005)
10. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: OntoViews - A Tool for Creating Semantic Web Portals. In: Proceedings of the Third Internation Semantic Web Conference, Springer Verlag (2004)
11. Reynolds, D., Shabajee, P., Cayzer, S.: Semantic Information Portals. In: Proceedings of the 13th International World Wide Web Conference on Alternate track papers & posters, ACM Press (2004)
12. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: Museumfinland – finnish museums on the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web **3** (2005) 224–241
13. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. (2005)
14. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: Semantic yellow page service discovery: The veturi portal. In: Poster paper, 4th International Semantic Web Conference. (2005)
15. Karger, D.R., Bakshi, K., Huynh, D., Quan, D., Sinha, V.: Haystack: A general-purpose information management tool for end users based on semistructured data. In: Proceedings of the CIDR Conference. (2005) 13–26
16. Quan, D., Huynh, D., Karger, D.R.: Haystack: A platform for authoring end user semantic web applications. In: Proceedings of the Second International Semantic Web Conference. (2003) 738–753
17. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: Proceedings of the Conference on Human Factors in Computing Systems, CHI. (2004) 415–422
18. Athanasis, N., Christophides, V., Kotzinos, D.: Generating on the fly queries for the semantic web: The ics-forth graphical rql interface (grql). In: Proceedings of the Third International Semantic Web Conference. (2004) 486–501
19. Catarci, T., Dongilli, P., Mascio, T.D., Franconi, E., Santucci, G., Tessaris, S.: An ontology based visual tool for query formulation support. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, IOS Press (2004) 308–312
20. Zhang, L., Yu, Y., Zhou, J., Lin, C., Yang, Y.: An enhanced model for searching in semantic portals. In: WWW ’05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, ACM Press (2005) 453–462
21. Fikes, R., Hayes, P., Horrocks, I.: Owl-ql: A language for deductive query answering on the semantic web. Technical report, Knowledge Systems Laboratory, Stanford University, Stanford, CA (2003)
22. Anyanwu, K., Sheth, A.P.: ρ-queries: enabling querying for semantic associations on the semantic web. In: Proceedings of the 12th international conference on World Wide Web. (2003) 690–699
23. Singh, S., Dey, L., Abulaish, M.: A framework for extending fuzzy description logic to ontology based document processing. In: Advances in Web Intelligence, Proceedings of the Second International Atlantic Web Intelligence Conference. (2004) 95–104

24. Parry, D.: A fuzzy ontology for medical document retrieval. In: CRPIT '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2004) 121–126
25. Kauppinen, T., Hyvönen, E. In: Modeling and Reasoning about Changes in Ontology Time Series. Springer-Verlag (2005) In press.