Searching and Ranking Ontologies on the Semantic Web

Edward Thomas
Computer Science Dept.
Aberdeen University
Aberdeen, UK
ethomas@csd.abdn.ac.uk

Harith Alani
Dept. of Electronics and
Computer Science
Uni. of Southampton
Southampton, UK
h.alani@ecs.soton.ac.uk

Derek Sleeman
Computer Science Dept.
Aberdeen University
Aberdeen, UK
sleeman@csd.abdn.ac.uk

Christopher Brewster
Computer Science Dept.
Uni. of Sheffield,
Sheffield, UK
C.Brewster@dcs.shef.ac.uk

ABSTRACT

The number of ontologies available online is increasing constantly. Tools that are capable of searching, retrieving, and ranking ontologies are becoming crucial to facilitate ontology search and reuse. In this document, we describe OntoSearch, which is a tool for capturing and searching ontologies on the Semantic web. We also briefly describe AKTiveRank which is used to rank OWL ontologies based on certain ontology-structure analysis.

Categories and Subject Descriptors

I.2.4 Knowledge Representation Formalisms and Methods – *representation languages, semantic networks.* H.3.3 Information Search and Retrieval – *Search process*, *Selection process*.

Keywords

Ontologies, Semantic Web, OWL, Search Engine

1. INTRODUCTION AND MOTIVATION

Finding a suitable ontology online is a hard task because of the difficulty of separating ontology schema from the mass of instance data and quickly evaluating its suitability.

There is still no good tool to handle this problem. Google offers a powerful web search engine. However, with regard to ontology searching, it has its own problems, such as a lack of visualisation facilities and poor summary information. Swoogle [3] provides a focussed search of ontologies on the semantic web, searching for specific keywords appearing as class or property names, but the search does not allow other properties of an ontology such as structure to be searched.

An opportunity was identified for a tool which provides the breadth of search possible through Google, along with additional functionality to help users visualise these results.

OntoSearch allows the user to specify different types of criteria (see next section) and returns a number of ontologies which match these criteria for the user to visualise and evaluate. The search results are then interpreted by AKTiveRank, which ranks the ontologies using the original search criteria.

2. ONTOSEARCH

OntoSearch[1][2] has grown from a system which used the Google API and provided additional filtering and information on the results returned to a hybrid system which searches a local repository and only reverts to Google when it does not have local information.. This functionality was developed to fulfil several requirements defined during user evaluations.

- The ability to specify the type of file(s) to be returned (OWL, RDF, all)
- The ability to specify the type of entities to be matched by each keyword (concept, attribute, values, comments, all)
- The ability to specify partial or exact matches on entities. So in partial match mode CHEMICAL would match CHEMICALS, CHEMICAL_AGENTS, etc; and of course in exact matching mode, only CHEMICAL would be matched.
- The ability to specify a sub-graph to be searched for. For example, concept Animal with concept Pig within 3 links; animals with particular attributes would be a further variant.

This required the implementation of a more advanced architecture with a triple store to provide a repository of Ontological information.

Two search strategies are currently possible using OntoSearch. Searching for structure using a simple query language which allows all the requirements identified to be covered or searching for classes using a keyword based search which is currently more restrictive.

2.1 Searching Structure

Structure based searching uses a simple query representation to describe subgraphs which might be present in an

K-CAP'05, October 2-5, 2005, Banff, Canada.
Copyright 2005 ACM 1-58113-000-0/00/0000…$5.00

ontology. It uses a query format based on N/Triples. A query fragment to describe a class called "animal" with a subclass called "cat" is shown below:

```
<"animal"=>$a> <22-rdf-syntax-ns#type> <owl#Class>
<"cat"> <rdfs#subClassOf> <$a>
```

Figure 1. Query Fragment

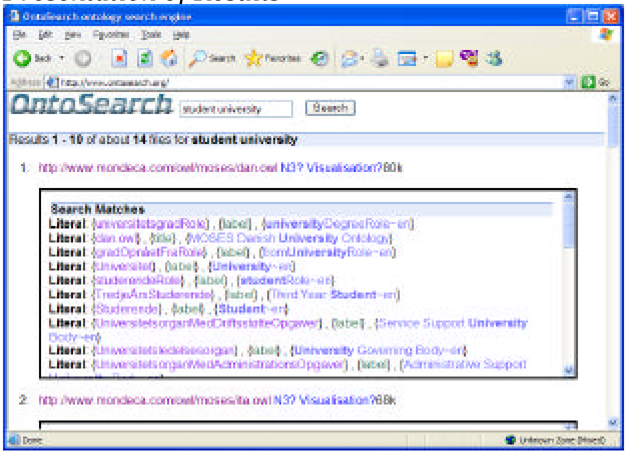
Queries are constructed from several of these fragments, using variables assigned with dollar signs to link each fragment and construct structural searches. Matching ontologies are returned to the user. We plan to enhance this functionality with a new formalism for searching ontologies, along with an enhanced visualisation tool and web service interface which will allow matching subgraphs to be highlighted and presented to the user so that the suitability of each match can be more easily evaluated.

2.2 Searching Classes

Class based searching uses keywords to match class information. The query searches for matches in class names, labels and comments and can match ontologies which contain any/all the query terms. Keywords and other information entered into the search form are used to create a query in the language used for all other OntoSearch queries.

The system returns a reference for each ontology which matched the search criteria, as well as the URI of each matching class within that ontology.

Presentation of Results



OntoSearch student university Search

Results 1 - 10 of about 14 files for **student university**

1. http://www.mondeca.com/owl/moses/dan.owl N3? Visualisation? 80k

Search Matches
Literal {universitetsgradRole} , {label} , {universityDegreeRole~en}
Literal {dan.owl} , {title} , {MOSES Danish University Ontology}
Literal {gradOpnaetFraRole} , {label} , {fromUniversityRole~en}
Literal {Universitet} , {label} , {University~en}
Literal {studerendeRole} , {label} , {studentRole~en}
Literal {TredjeAarsStuderende} , {label} , {Third Year Student~en}
Literal {Studerende} , {label} , {Student~en}
Literal {UniversitetsorganMedDriftsstatteOpgaver} , {label} , {Service Support University Body~en}
Literal {Universitetsledelsesorgan} , {label} , {University Governing Body~en}
Literal {UniversitetsorganMedAdministrationsOpgaver} , {label} , {Administrative Support

2. http://www.mondeca.com/owl/moses/ita.owl N3? Visualisation? 68k

Figure 2. Presentation of results

Results in HTML present information to the user that should make evaluation of the results as straight forward as possible (figure 2). Results in RDF results allow the system to be used as a Proto-Web-Service with results being used by an external system to provide additional functionality. A simple TouchGraph[1] based visualisation (figure 3) of the class structure of the ontology is available.

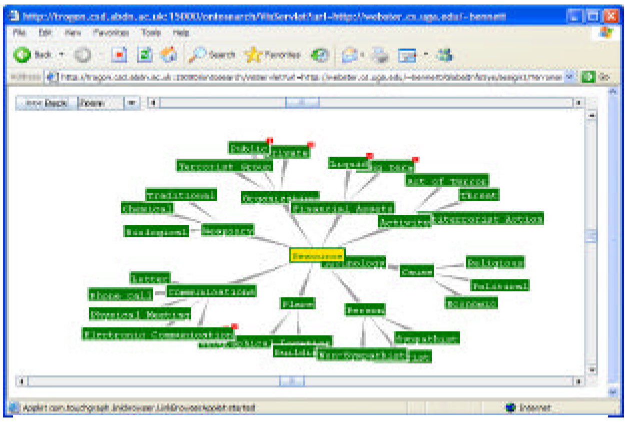


Figure 3. TouchGraph Visualisation

The results from the class search is a list of Ontologies and the URIs of all the classes which matched one of more of the keywords searched for. The RDF file is used by AKTiveRank to rank the matched ontologies returned by OntoSearch against the search terms.

2.3 Implementation

OntoSearch is based around a custom built triple store, based on Berkeley DB Database2 technology. This optimizes the fast querying of ontological data, and it forms the core repository which the user queries are evaluated against. To provide as wide a search as possible, each query is first processed to extract keywords; in the case of a structure query this is any literal value included in the query terms (in the example fragment given above, this would be "animal" and "cat"), in the case of a class search these are the same as the keywords entered by the user. These keywords are used to build a Google query using techniques refined in previous versions of OntoSearch to return only ontological data from the Semantic Web. The Google query is executed through the Google API and the first 100 results (or all results if fewer files are returned) are examined and if not currently present in the repository, they are downloaded and added to it. This allows us to have access to the largest possible database of ontological data on the Semantic Web and offer a wider search than possible if we were using our own spider.

[1] TouchGraph: www.touchgraph.com

[2] Berkeley DB: www.sleepycat.com

The repository stores a list of all queries made to Google, additionally it records the date and time the query was made and the files retained. As each query is evaluated by the system, this list is checked to see if the repository needs to be updated with a new Google query. If an identical query (when broken down into Google keywords) has been made on Google within the last 7 days, then no new query is made and the repository is queried as is. If Google has not been queried with these keywords recently, then a new Google query is made and the first 100 results are examined, only those results which refer to files which are either not present in the repository or whose files are more than 7 days old in the repository are downloaded again. This means that after an initial query in a specific domain, all subsequent queries are much quicker as at least a part of the search will take place locally in the OntoSearch repository.

Once the repository has been updated, the query is performed on the database. All URI references and keywords are looked up in an index of RDF resources, returning either unique values which are used to represent the URI in the triple store or a list of keyword matches. Each RDF triple in the database is made up of three references to the RDF resources which are used in the database.

This data is combined with the original query and compiled into a Java object which can be reused for subsequent queries. The query engine uses several metrics based on the statistical nature of the database to ensure that each query statement is executed in the optimal order for best performance; this allows to perform ontology search within the database (excluding Google search and compilation of cached data in the repository) in under a second for most queries.

3. ONTOLOGY RANKING

Ranking ontologies is an important issue, especially when many potentially-relevant ontologies are found. Swoogle [3] and OntoKhoj [4] rank ontologies using a PageRank [5] method that analyses links and referrals between ontologies to identify the most popular ontologies. However, the majority of ontologies available on the Web are poorly connected, and more than half of them are not referred to by any other ontology [3], which will likely produce poor PageRank results.

Furthermore, a popular ontology does not necessarily indicate a good representation of all the concepts it covers [6]. For example, suppose a user was looking for an ontology about "students"; there could be an ontology about the general academic domain that is well connected, and thus popular. If this ontology contains a concept named "Student", then this ontology will show up high on the list of candidates. However, it could very well be the case that the "Student" class is very weakly represented in this particular ontology. This ontology might have become popular due to its coverage of conferences and research topics, rather than for its coverage of more student related concepts.

3.1 AKTiveRank

AKTiveRank [6] is a prototype system for ranking ontologies by aggregating a number of graph-analysis measures that use certain structural features of concepts, such as their hierarchical centrality, structural density, and semantic similarity to other concepts of interest.

3.2 Implementation

AKTiveRank applies four types of assessments (measures) for each ontology to measure the rankings:

1. **Class Match Measure (CMM):** Evaluates the coverage of an ontology for the given search terms. An ontology that contains all search terms will obviously score higher than others, and exact matches are regarded as better than partial matches.
2. **Centrality Measure (CEM):** Studies showed that mid-hierarchical-level concepts tend to be more detailed and prototypical of their categories than classes at higher or lower hierarchical levels [7]. CEM measures how close a concept is placed to the middle level of its hierarchy.
3. **Density Measure (DEM):** When searching for a "good" representation of a specific concept, one would expect to find a certain degree of detail in the representation for the target concept. This may include how well the concept is further specified, how many attributes and siblings the class has, etc. DEM is intended to approximate the representational-density of classes and consequently the level of detail for concepts.
4. **Semantic Similarity Measure (SSM):** This measure calculates the semantic similarity between the classes that were matched in the ontology with the search terms. The motivation here is that it might be preferred for the search terms to be closely related to each other in the ontology than otherwise. SSM formula based on the shortest path measure defined in [8].

The Total Score of an ontology is calculated once the four measures are applied to all the returned ontologies. Total score is the aggregation of all the measures' values, taking into account their *weights*, which are used to determine the importance of each measure in the ranking.

4. RESULTS

To evaluate the utility of OntoSearch/AKTive Rank, we performed queries for the concepts "Student" and "University". OntoSearch was queried in ClassName mode, searching for OWL ontologies containing matches for every keyword in the query. The results as RDF were passed to AKTiveRank for ranking.

4.1 OntoSearch Results

At the time of writing, OntoSearch returned the following ontology URIs for the query; Student University:

http://www.mondeca.com/owl/moses/dan.owl
http://www.mondeca.com/owl/moses/ita.owl

http://www.lehigh.edu/~zhp2/univ-bench.owl
http://www.architexturez.net/sub.gate/metadata/onto-caad/caad.ka.n3.owl
http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE_inst.owl
http://protege.stanford.edu/plugins/owl/owl-library/ka.owl
http://ontology.deri.org/docs/swportal.owl
http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies/russia2.owl

4.2 AKTiveRank Results

OntoSearch returned information about eight ontologies, but as information about 2 of the ontologies was unavailable online at the time, they were dropped from the ranking process. So AKTive Ranking ranked the remaining 6 ontologies (namely univ-bench, russia2, dan, ita, ka and swportal) as shown in figure 4. It can be seen that the *univ-bench* ontology scored overall highest. The algorithm calculated the AKTiveRank scores by applying weighting factors for each measure. Here we used the weights 0.6, 0.4. 0.8 & 0.7 respectively for the corresponding CEM, CMM, DEM & SSM measures for each of the ontologies. The composite score for each of the ontologies is also included in Figure 4. The *univ-bench* ontology scored higher than any of the others in all measures, except for SSM, where the *russia2* ontology scored the highest value. Further tests are required to identify the optimal set of weights for aggregating the four ranking measures currently used. This will require further human-based ranking study [6].

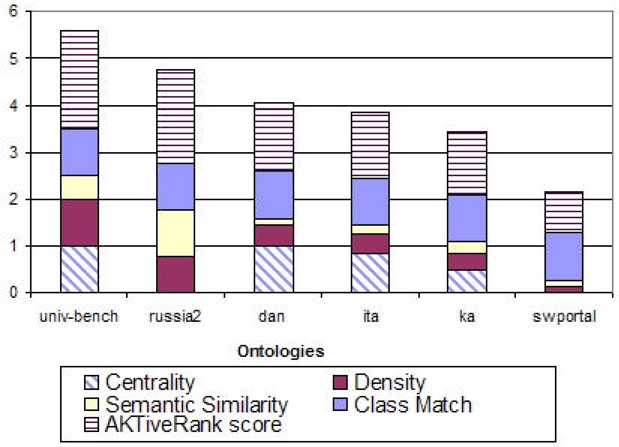


Figure 4: AKTiveRank results

5. FURTHER WORK

5.1 OntoSearch

As the repository of ontological data grows, this gives us a larger database to conduct queries of this data. Yi Zhang is currently working on applying query refinement techniques to ontology searching, which will help the user to clearly specify his/her knowledge requirements and further to express them into advanced queries. This work will be integrated with OntoSearch. The current keyword interface will be supplemented with a query builder to allow users to work with this query language and build effective queries. There will also be a more advanced visualisation system to allow the specific fragments of an ontology which match a query to be highlighted and explored in more detail than is currently possible with the OntoSearch visualisation tool. We will also expand the current API to comply with the W3C definition of a web service, allowing the functionality provided by OntoSearch to be integrated into other applications.

Another extension of OntoSearch currently in development is the ability for users to submit sites to be added to the repository as well as those available through Google. This will allow OntoSearch to index and access sources which do not get indexed by Google or do not contain terms which directly match the Google searches performed by OntoSearch. Sites submitted into this database will be spidered and indexed at regular intervals to ensure that the repository is kept up to date.

5.2 AKTive Rank

The parameters used in the AKTive Rank process need to be reconsidered in the light of the needs of human knowledge engineers. In order to do this, we plan a more extensive human ranking study which will include a larger population of subjects and will try to elicit a greater understanding of the process of ontology evaluation and selection.

Another problem is the inadequacy of existing RDF query languages in dealing with graph queries, such as those required in SSM. We are in the process of moving such queries to JUNG[3], which is a better graph querying systems.

ACKNOWLEDGMENTS

This work is supported under the Advanced Knowledge Technologies (AKT) IRC (EPSRC grant no. GR/N15764/01) comprising Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

REFERENCES

[1] Zhang Y, Vasconcelos W, and Sleeman D. OntoSearch: An Ontology Search Engine (AI-2004). The 24th SGAI Int. Conf. on Innovative Techniques and Applications of AI, Cambridge, 2004.

[2] Thomas E, Zhang Y, Sleeman D, Preece A, McKenzie C and Wright J. OntoSearch: a Service to Support the Reuse of Ontologies. Demos and Posters of the 2nd European Semantic Web Conference (ESWC), 2005.

[3] Ding L, Finin T, Joshi A, Pan R, Cost R.S, Peng Y, Reddivari P, Doshi V.C, and Sachs J. Swoogle: A semantic

[3] http://jung.sourceforge.net/

web search and metadata engine. In Proc. 13th ACM Conf. on Info. & Knowledge Management, 2004.

[4] Patel C, Supekar K, Lee Y, and Park E. Ontokhoj: A semantic web portal for ontology searching, ranking, and classification. In Proc. 5th ACM Int. Workshop on Web Information and Data Management, pages 58–61, New Orleans, Louisiana, USA, 2003.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.

[6] Alani H, Brewster C. Ontology Ranking based on the Analysis of Concept Structures. Third Int. Conf. on Knowledge Capture (K-Cap), Banff, 2005.

[7] Rosch E. Principles of Categorization. in E. Rosch and B. B. Lloyd editors. Cognition and Categorization, Lawrence Erlbaum, Hillsdale, New Jersey, 1978.

[8] Rada R, Mili H, Bicknell E, Blettner M. Development and application of a metric on semantic nets. IEEE Trans. Sys. Man. and Cybernetics, 19(1):17–30, 1989.