

# Ranking Ontologies with AKTiveRank

Harith Alani<sup>1</sup>, Christopher Brewster<sup>2</sup>, and Nigel Shadbolt<sup>1</sup>

<sup>1</sup> Intelligence, Agents, Multimedia  
School of Electronics and Computer Science  
University of Southampton, Southampton, UK  
{h.alani, nrs}@ecs.soton.ac.uk

<sup>2</sup> Dept. of Computer Science  
University of Sheffield, Sheffield, UK  
C.Brewster@dcs.shef.ac.uk

**Abstract.** Ontology search and reuse is becoming increasingly important as the quest for methods to reduce the cost of constructing such knowledge structures continues. A number of ontology libraries and search engines are coming to existence to facilitate locating and retrieving potentially relevant ontologies. The number of ontologies available for reuse is steadily growing, and so is the need for methods to evaluate and rank existing ontologies in terms of their relevance to the needs of the knowledge engineer. This paper presents AKTiveRank, a prototype system for ranking ontologies based on a number of structural metrics.

## 1 Introduction

Knowledge representation in the Semantic Web will be largely based on ontologies. However, ontology construction remains challenging, mainly due to the skill, time, effort, and domain specific knowledge required. In order to minimise this, one of the major advantages claimed of ontologies is the potential of “reuse”. Publicly available ontologies are to be reused, modified, extended, and pruned as required, thereby avoiding the huge effort of starting from scratch [1].

Search engines to help finding relevant ontologies have started to appear in recent years. Swoogle<sup>3</sup> [6] is currently dominating this area of development, indexing an increasing number of ontologies covering a wide range of domains.

As the number of ontologies that such search engines can find increases, so will the need increase for a proper ranking method to order the returned lists of ontologies in terms of their relevancy to the query. This could save a lot of time and effort by reducing the need to examine in detail each and every ontology returned to find out how well it suits the needs of the agent or knowledge engineer.

Evaluating and ranking ontologies can be based on many different criteria [9]. This paper presents AKTiveRank, a prototype of an ontology ranking system which applies a number of analytic methods to rate each ontology based on an estimation of how well it represents the given search terms. AKTiveRank could be integrated with other, different, ranking systems to include additional ranking criteria, such as user ratings or content coverage.

---

<sup>3</sup> <http://swoogle.umbc.edu/>

Related work concerning ontology evaluation and ranking is reviewed in the following section. A full description of the architecture and ranking method is given in section 3. An experiment is detailed in section 4 and evaluated in section 5. Future work and Conclusions are discussed in the final sections of the paper.

## 2 Ontology Evaluation and Ranking

Lack of automatic, well grounded, methodologies to evaluate ontologies may seriously hinder their adoption by the industry and the wider web community [9]. Ontologies may be assessed from different angles, such as how the ontologies have been rated and reviewed by users (e.g. [22]), how well they meet the requirements of certain evaluation tests (e.g. [11]) or general ontological properties (e.g. [14]).

Gangemi and colleagues [9] define three main types of evaluation; functional, usability-based, and structural evaluation. Functional evaluation focuses on measuring how well an ontology is serving its purpose (e.g. [4]). Usability evaluations is concerned with metadata and annotations (e.g. [10]). Structural evaluation focuses on the structural properties of the ontology as a graph (e.g. [2]).

Other criteria for evaluating an ontology can be based on its content coverage. Jones and Alani are experimenting with ranking ontologies based on a tf/idf comparison of each potentially relevant ontology with an automatically gathered corpus that describes the domain of interest [12].

Some ontology search engines adopted a PageRank-like method to rank ontologies by analysing links and referrals between the ontologies in the hope of identifying the most popular ones (e.g. Swoogle [6, 7] and OntoKhoj [17]). However, this method of ranking will not work for a large number of existing ontologies because of their poor connectivity and lack of referrals from other ontologies [6]. Such ‘self contained’ or ‘isolated’ ontologies would certainly receive poor PageRank results, thus highlighting the need for additional ranking methods. Furthermore, the *popularity* of an ontology could be a good overall assessment of the ontology, but it does not necessarily correlate with ‘good’ or appropriate representations of specific pieces of knowledge (e.g. certain classes) [2].

Based on the various evaluation and ranking methods mentioned above, it is clear that there is a need to assess all important features of an ontology. This can provide a multi-dimensional ranking approach that users can control as required.

AKTiveRank is an experimental system for ranking ontologies based on a number of measures that assess the ontology in terms of how well it represents the concepts of interest. Users are assumed to be using an ontology search engine (e.g. Swoogle) to do the search. The query submitted to the search engine is used by AKTiveRank to identify the concepts that match the user’s request. The ranking measures applied by AKTiveRank will be based on the representation of those concepts and their neighbourhoods.

This paper experiments with a modified set of ranking measures to those we previously used and described in [2]. The measures and an experiment are presented in the following sections.

### 3 AKTiveRank

Figure 1 shows the current architecture of AKTiveRank. The main component (number 2 in the figure) is a Java Servlet that receives an HTTP query from a user or an agent (no. 1). The query contains the terms to search for. Currently it is only possible to search for concepts. In other words, search terms will only be matched with ontology classes, and not with properties or comments. This is simply to focus AKTiveRank on assessing the representation of the concepts of interest to the user.

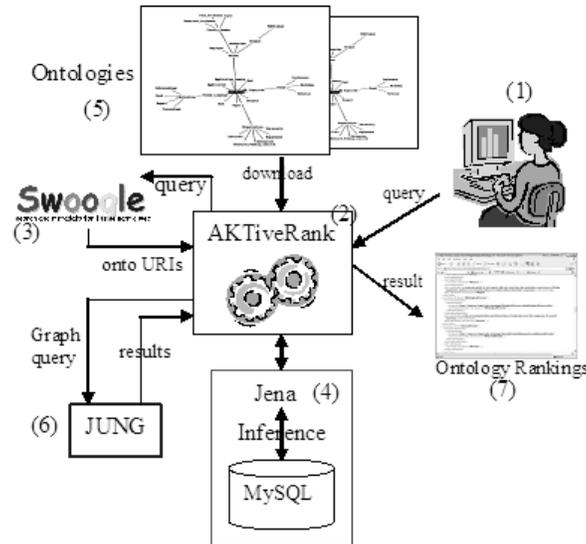


Fig. 1. AKTiveRank Architecture

When a search query is received, AKTiveRank forwards the query to Swoogle (no. 3) and retrieves the returned ontology URIs. Even though AKTiveRank currently relies on Swoogle to get the list of potentially relevant ontologies to be ranked, it is in no way restricted to it. Other sources and methods for searching for ontologies can also be used to feed AKTiveRank with lists of ontologies.

Once a list of ontology candidates is gathered, AKTiveRank starts to check whether those ontologies are already stored in a Jena MySQL database back-end (no. 4), and if not, downloads them from the web (no. 5) and add them to the database. The Jena API is used here to read the ontologies and handle the database storage.

Existing RDF query languages are not well suited for graph queries. To this end, the current version of AKTiveRank is connected to a purpose-built JUNG servlet (no. 6), which receives an ontology URI and sends back results of JUNG queries in RDF. JUNG (Java Universal Network/Graph framework) is a software library for analysing and visualising network graphs.

AKTiveRank then analyses each of the ontology candidates to determine which is most relevant to the given search terms. This analysis will produce a ranking of the retrieved ontologies, and the results are returned to the user as an OWL file containing the ontology URIs and their total ranks.

### 3.1 The Ranking Measures

AKTiveRank applies four measures to evaluate different representational aspects of the ontology and calculate its ranking. Each ontology is examined separately. Once those measures are all calculated for an ontology, the resulting values will be merged to produce the total rank for the ontology.

The measures used in AKTiveRank are experimental and subject to change. In a previous version of AKTiveRank which was reported in [2], one of the measures applied was the Centrality Measure (CEM). That measure aimed to assess how representative a class is of an ontology based on the observation that the closer a class is to the middle level of the hierarchy, the more likely it is that the representation of the class is well detailed [20]. However, in some experiments we found a few ontologies that placed our concept of interest near the top of the hierarchy. Those few ontologies were entirely focused around the concept we were searching for. This meant that even though such ontologies can be highly relevant to our search, they scored very low in CEM. Furthermore, we also found that CEM values corresponded in most cases to the values of the Density measure, and rendered CEM somewhat redundant.

The new implementation of AKTiveRank also introduces a new measure; the Betweenness measure, and extends the Class Match measure as described in the following sections. An example on calculating the values for an ontology will be given for each of the four measures currently used by AKTiveRank.

**Class Match Measure** The Class Match Measure (CMM) is meant to evaluate the coverage of an ontology for the given search terms. Similar metrics have been used in the past as part of measures to estimate similarity of software descriptions [21].

AKTiveRank looks for classes in each ontology that have labels matching a search term either exactly (class label “identical to” search term) or partially (class label “contains” the search term). An ontology that covers all search terms will obviously score higher than others, and exact matches are regarded as better than partial matches. For example if searching for “Student” and “University”, then an ontology with two classes labelled exactly as the search terms will score higher in this measure than another ontology which contains partially matching classes, e.g. “UniversityBuilding” and “PhD-Student” (see example below).

This measure has been extended from its previous version used in [2] by allowing it to take into account the total number of partially matching classes. In other words, if we are interested in the concept “student”, then the CMM value for this ontology will be higher the more classes it has with the given word appearing in their labels or URIs. In another study we found that taking partial matches into account can sometimes be problematic and may reduce the search quality [12] (e.g. “gene” and “generator”). Therefore, the use of partially matching class labels has been limited to CMM only for the time being. Only if an exact match is unavailable that a partial match is considered in the other three measures.

**Definition 1.** Let  $C[o]$  be a set of classes in ontology  $o$ , and  $T$  is the set of search terms.

$$E(o, T) = \sum_{c \in C[o]} \sum_{t \in T} I(c, t) \quad (1)$$

$$I(c, t) = \begin{cases} 1 & : \text{if } label(c) = t \\ 0 & : \text{if } label(c) \neq t \end{cases} \quad (2)$$

$$P(o, T) = \sum_{c \in C[o]} \sum_{t \in T} J(c, t) \quad (3)$$

$$J(c, t) = \begin{cases} 1 & : \text{if } label(c) \text{ contains } t \\ 0 & : \text{if } label(c) \text{ not contain } t \end{cases} \quad (4)$$

where  $E(o, T)$  and  $P(o, T)$  are the number of classes of ontology  $o$  that have labels that match any of the search terms  $t$  exactly or partially, respectively.

$$CMM(o, \tau) = \alpha E(o, T) + \beta P(o, T) \quad (5)$$

where  $CMM(o, \tau)$  is the Class Match Measure for ontology  $o$  with respect to search terms  $\tau$ .  $\alpha$  and  $\beta$  are the exact matching and partial matching weight factors respectively. Exact matching is favoured over partial matching if  $\alpha > \beta$ . In the experiments described in this paper,  $\alpha = 0.6$  &  $\beta = 0.4$ , thus putting more emphasis on exact matching.

*Example:* When searching the ontology  $o$  (aktive-portal-ontology-latest.owl<sup>4</sup>) for class labels that equals, or contains, the terms “student” or “university”, the following classes can be found: *Student, PhD-Student, University, Distance-teaching-university* and *University-faculty*. So the results is two classes with identical labels to our search terms, and three classes with labels containing the search terms. CMM can therefore be calculated as follows:

$$\begin{aligned} cmm(student) &= 1 * 0.6 + 1 * 0.4 = 1 \\ cmm(university) &= 1 * 0.6 + 2 * 0.4 = 1.4 \\ CMM(o, \{student, university\}) &= 1 + 1.4 = 2.4 \end{aligned}$$

**Density Measure** When searching for a specific concept, one would expect to find a certain degree of detail in the representation of the knowledge concerning that concept (i.e. a rich conceptual neighbourhood). This may include how well the concept is further specified (the number of subclasses), the number of properties associated with that concept, number of siblings, etc. All this is taken into account in the Density Measure (DEM). DEM is intended to approximate the representational-density or information-content of classes and consequently the level of knowledge detail.

Density calculations are currently limited to the numbers of *direct* relations, subclasses, superclasses, and siblings. We dropped the number of instances from this measure as this might skew the results unfairly towards populated ontologies which may not necessarily reflect the quality of the schema itself.

**Definition 2.** Let  $S = \{S_1, S_2, S_3, S_4\} = \{subclasses[c], superclasses[c], relations[c], siblings[c]\}$

<sup>4</sup> <http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl>

$$dem(c) = \sum_{i=1}^4 w_i |S_i| \quad (6)$$

$$DEM(o) = \frac{1}{n} \sum_{i=1}^n dem(c) \quad (7)$$

where  $dem(c)$  is the Density Measure for class  $c$ .  $w_i$  is a weight factor set to a default value of 1, and  $n = E(o, T) + P(o, T)$  which is the number of matched classes in ontology  $o$ .

*Example:* The neighbourhoods of the classes *Student* and *University* in the ontology ita.owl<sup>5</sup> are shown in figure 2. When using the weights 1, 0.25, 0.5 and 0.5, for subclasses, superclasses, relationships and siblings respectively, we get the following:

$$\begin{aligned} dem(student) &= 1 * 2 + 0.25 * 1 + 0.5 * 0 + 0.5 * 1 = 2.75 \\ dem(university) &= 1 * 0 + 0.25 * 1 + 0.5 * 0 + 0.5 * 5 = 2.75 \\ DEM(ita.owl) &= \frac{2.75 + 2.75}{2} = 2.75 \end{aligned}$$

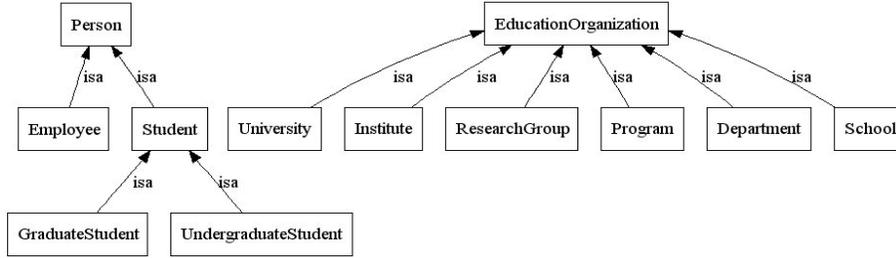


Fig. 2. Neighbourhood of *Student* and *University* in ita.owl ontology.

**Semantic Similarity Measure** Similarity measures have often been frequently explored in information retrieval systems to provide better ranking for query results (e.g. [5, 18]). Ontologies can be viewed as semantic graphs of concepts and relations, and hence similarity measures can be applied to explore these conceptual graphs. Resnik applied a similarity measure to WordNet to resolve ambiguities [19]. The measure he used is based on the comparison of shared features, which was first proposed in [23]. Another common-feature based similarity is the shortest-path measure, introduced by Rada [18]. He argues that the more relationships objects have in common, the closer they will be in an ontology. Variations of these techniques have been used to measure similarity between whole ontology structures [15, 24].

The Semantic Similarity Measure (SSM) calculates how close are the concepts of interest laid out in the ontology structure. If the concepts are positioned relatively far from each others, then it becomes unlikely for those concepts to be represented in a compact manner, rendering their extraction and reuse more difficult. Further studies are required to find whether or not this assumption is dependent on certain ontological properties, such as size or level of detail.

<sup>5</sup> <http://www.mondeca.com/owl/moses/ita.owl>

The SSM formula used here is based on the simple shortest path measure defined in [18]. SSM is measured from the minimum number of links that connects a pair of concepts. These links can be isA relationships or other object properties.

**Definition 3.** Let  $c_i, c_j \in \{classes[o]\}$ , and  $c_i \xrightarrow{p} c_j$  is a path  $p \in P$  of paths between classes  $c_i$  and  $c_j$

$$ssm(c_i, c_j) = \begin{cases} \frac{1}{length(\min_{p \in P} \{c_i \xrightarrow{p} c_j\})} & : \text{if } i \neq j \\ 1 & : \text{if } i = j \end{cases} \quad (8)$$

$$SSM(o) = \frac{1}{k} \sum_{i=1}^{n-1} \sum_{j=i+1}^n ssm(c_i, c_j) \quad (9)$$

where  $n$  is the number of matched classes, and  $k = \sum_{k=1}^{n-1} k$ . Note that even though  $ssm(c_i, c_i) = 1$ , the system never actually needs to compare a class with itself.

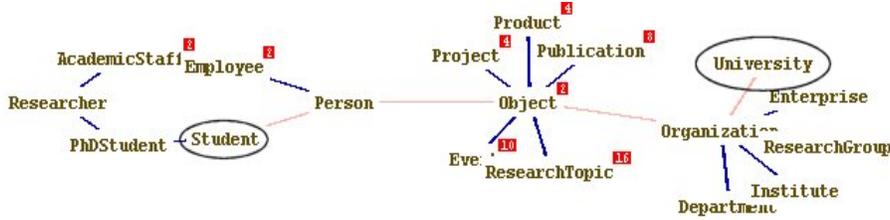


Fig. 3. Shortest path between *Student* and *University* in ka.owl ontology.

Example: Figure 3 shows the shortest path between the classes *Student* and *University* in the ka.owl<sup>6</sup> ontology. Applying SSM to these two classes will produce:

$$ssm(student, university) = \frac{1}{4}$$

$$SSM(ka.owl) = \frac{1}{1} ssm(student, university) = 0.25$$

**Betweenness Measure** One of the algorithms that JUNG provides is Betweenness [8]. This algorithm calculates the number of shortest paths that pass through each node in the graph. Nodes that occur on many shortest paths between other nodes have higher betweenness value than others. The assumption for using this measure in AKTiveRank is that if a class has a high betweenness value in an ontology then this class is *graphically central* to that ontology.

The BETweenness Measure (BEM) calculates the betweenness value of each queried concept in the given ontologies. The ontologies where those classes are more central will receive a higher BEM value.

**Definition 4.** Let  $c_i, c_j \in \{classes[o]\}$ ,  $c_i$  and  $c_j$  are any two classes in ontology  $o$ ,  $C[o]$  is the set of class in ontology  $o$ ,  $bem(c)$  is the BETweenness Measure for class  $c$ .

<sup>6</sup> <http://protege.stanford.edu/plugins/owl/owl-library/ka.owl>

$$bem(c) = \sum_{c_i \neq c_j \neq c \in C[o]} \frac{\sigma_{c_i c_j}(c)}{\sigma_{c_i c_j}} \quad (10)$$

where  $\sigma_{c_i c_j}$  is the shortest path from  $c_i$  to  $c_j$ , and  $\sigma_{c_i c_j}(c)$  is the number of shortest paths from  $c_i$  to  $c_j$  that passes through  $c$ .

$$BEM(o) = \frac{1}{n} \sum_{k=1}^n bem(c_k) \quad (11)$$

where  $n$  is the number of matched classes in ontology  $o$ , and  $BEM(o)$  is the average Betweenness value for ontology  $o$ .

*Example:* When BEM is applied to the classes *Student* and *University* of the *univ.owl*<sup>7</sup> ontology, the class *Student* received a value of 0.00468, while *University* got a 0 betweenness value (using the Betweenness Centrality measure of Jung). This means that the former class is more central in the ontology graph than the later class. Final BEM value can then be calculated as follows:

$$BEM(univ.owl) = \frac{1}{2}(0.00468 + 0.0) = 0.00234.$$

### 3.2 Total AKTiveRank Score

The total score of an ontology can be calculated once the four measures are applied to all the ontologies that the search engine returned. Total score is calculated by aggregating all the measures' values, taking into account the *weight* of each measure, which can be used to determine the relative importance of each measure for ranking.

The first rank will be given to the ontology with the highest overall score, the second rank to the second highest score, and so on.

**Definition 5.** Let  $M = \{M[1], \dots, M[i], M[4]\} = \{CMM, DEM, SSM, BEM\}$ ,  $w_i$  is a weight factor, and  $O$  is the set of ontologies to rank.

$$Score(o \in O) = \sum_{i=1}^4 w_i \frac{M[i]}{\max_{1 \leq j \leq |O|} M[j]} \quad (12)$$

Values of each measure are normalised to be in the range (0–1) by dividing by the maximum measure value for all ontologies. For example, if the maximum DEM value calculated for a set of ontologies is 4.75, then the normalised DEM value for the ontology *ita.owl* (example in sec. 3.1) will be  $\frac{2.75}{4.75} = 0.579$  (table 2).

## 4 Experiment

In this section we report the results of running AKTiveRank over an example query submitted to Swoogle<sup>8</sup>.

<sup>7</sup> <http://www.mondeca.com/owl/mones/univ.owl>

<sup>8</sup> Using Swoogle 2005

The weights for calculating total score (equation 12) for our experiment are set to 0.4,0.3,0.2,0.1 for the CMM, BEM, SSM, DEM measures respectively. The relative weights for these measures are selected based on how well each measure performed in our evaluation (section 5). Further tests are required to identify the best weights to use, and whether the chosen mix of weights applies equally well to other queries and ontologies.

**Table 1.** Order of search result for “student university” as returned by Swoogle. Duplicates were removed.

	Ontology URL
a	<a href="http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl">http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl</a>
b	<a href="http://protege.stanford.edu/plugins/owl/owl-library/koala.owl">http://protege.stanford.edu/plugins/owl/owl-library/koala.owl</a>
c	<a href="http://protege.stanford.edu/plugins/owl/owl-library/ka.owl">http://protege.stanford.edu/plugins/owl/owl-library/ka.owl</a>
d	<a href="http://reliant.teknowledge.com/DAML/Mid-level-ontology.owl">http://reliant.teknowledge.com/DAML/Mid-level-ontology.owl</a>
-	<a href="http://www.csee.umbc.edu/~shashi1/Ontologies/Student.owl">http://www.csee.umbc.edu/~shashi1/Ontologies/Student.owl</a>
e	<a href="http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl">http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl</a>
f	<a href="http://www.mondeca.com/owl/moses/univ2.owl">http://www.mondeca.com/owl/moses/univ2.owl</a>
g	<a href="http://www.mondeca.com/owl/moses/univ.owl">http://www.mondeca.com/owl/moses/univ.owl</a>
-	<a href="http://www.lehigh.edu/~yug2/Research/SemanticWeb/LUBM/University0_0.owl">http://www.lehigh.edu/~yug2/Research/SemanticWeb/LUBM/University0_0.owl</a>
h	<a href="http://www.lri.jur.uva.nl/~rinke/aargh.owl">http://www.lri.jur.uva.nl/~rinke/aargh.owl</a>
-	<a href="http://www.srdc.metu.edu.tr/~yildiray/HW3.OWL">http://www.srdc.metu.edu.tr/~yildiray/HW3.OWL</a>
i	<a href="http://www.mondeca.com/owl/moses/ita.owl">http://www.mondeca.com/owl/moses/ita.owl</a>
j	<a href="http://triplestore.aktors.org/data/portal.owl">http://triplestore.aktors.org/data/portal.owl</a>
k	<a href="http://annotation.semanticweb.org/ontologies/iswc.owl">http://annotation.semanticweb.org/ontologies/iswc.owl</a>
-	<a href="http://www.csd.abdn.ac.uk/cmckenzi/playpen/rdf/abdn_ontology_LITE.owl">http://www.csd.abdn.ac.uk/cmckenzi/playpen/rdf/abdn_ontology_LITE.owl</a>
l	<a href="http://ontoware.org/frs/download.php/18/semiport.owl">http://ontoware.org/frs/download.php/18/semiport.owl</a>

Now let's assume that we need to find an OWL ontology that represents the concepts of “University” and “Student”. The domain of academia is good for such experiments due to the relatively large number of relevant ontologies about this domain. The list of ontologies returned by Swoogle at the time of the experiment as a result of the query “university student type:owl” is shown in table 1. Some of those ontologies were duplicates (i.e. the same ontology was available under two slightly different URLs). As expected, the same rank was produced by AKTiveRank for all duplicated ontologies, and therefore were removed from the table to reduce complexity.

Some ontologies were no longer online and hence were dropped from the ranking experiment (they are given index “-” in the table).

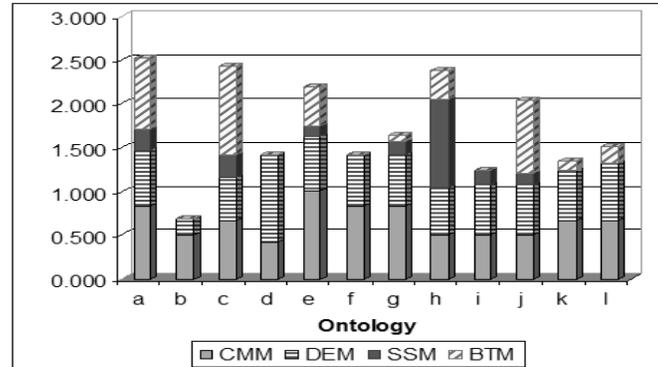
When AKTiveRank was applied to the results list in table 1, it produced the values given in table 2. Figure 4 shows the sum of results per ontology, without applying any measure weightings. These values are obtained by calculating the values for each of AKTiveRank’s measure, then normalising each value by dividing it by the maximum value calculated for that measure when applied to all identified ontologies.

#### 4.1 Results Analysis

From the results of this experiment, it can be seen that ontology *a* scored the highest value in AKTiveRank. The ontologies *c* and *h* were given the second and third rank respectively. The *koala* ontology, which was placed second in Swoogle’s results list, got the least AKTiveRank score, and thus was placed last in the ranked list. Even though this ontology contains classes labelled “Student” and “University”, those classes are not

**Table 2.** Normalised AKTiveRank results. Results for each ontology are weighted and aggregating to produce a final score, which is compared with the other scores to set the rank

Onto	CMM	DEM	SSM	BEM	Score	Rank
a	0.833	0.632	0.250	0.806	0.688	1
b	0.5	0.197	0	0	0.220	12
c	0.667	0.5	0.25	1	0.667	2
d	0.417	1	0	0	0.267	11
e	1	0.632	0.111	0.452	0.621	3
f	0.833	0.579	0	0	0.391	7.5
g	0.833	0.579	0.167	0.065	0.444	6
h	0.5	0.553	1	0.323	0.552	4
i	0.5	0.579	0.167	0	0.291	10
j	0.5	0.579	0.125	0.839	0.535	5
k	0.667	0.579	0	0.097	0.354	9
l	0.667	0.685	0	0.194	0.391	7.5



**Fig. 4.** Aggregated AKTiveRank scores using equal weights

closely associated (i.e. zero SSM<sup>9</sup>) and not graphically central to the ontology structure (i.e. zero BEM). The *koala* ontology is not exactly about students or universities, and therefore deserves the last rank in this context.

Note that 5 of our ontologies received a SSM of 0.0. This indicates that AKTiveRank did not manage to find any paths connecting the two given queried classes. Semantic paths that cross via *owl:Thing* class are ignored.

The ontology that scored the highest in the Class Match measure (CMM, section 3.1) was ontology *e*. This ontology had 2 classes with labels exactly matching our search terms, and 3 partially matching ones; *Phd-Student*, *University-Faculty* and *Distance-Teaching-University*.

The highest DEM value was calculated for ontology *d*. This ontology had a total of 5 subclasses and 10 siblings for the two classes matching our search terms. This added to its DEM value and made this ontology score best on this measure.

Ontology *h* received the maximum SSM value because it has the relation *enrolled\_at* which directly connects the classes “Student” and “University”.

And finally, ontology *c* was found to have the highest average betweenness value for the two classes in question, which indicates that these classes are more structurally central in this ontology than in the other ontologies.

<sup>9</sup> Jena disagrees with Proègè OWL on its rendering of a restriction in the Koala ontology between the classes Student and University.

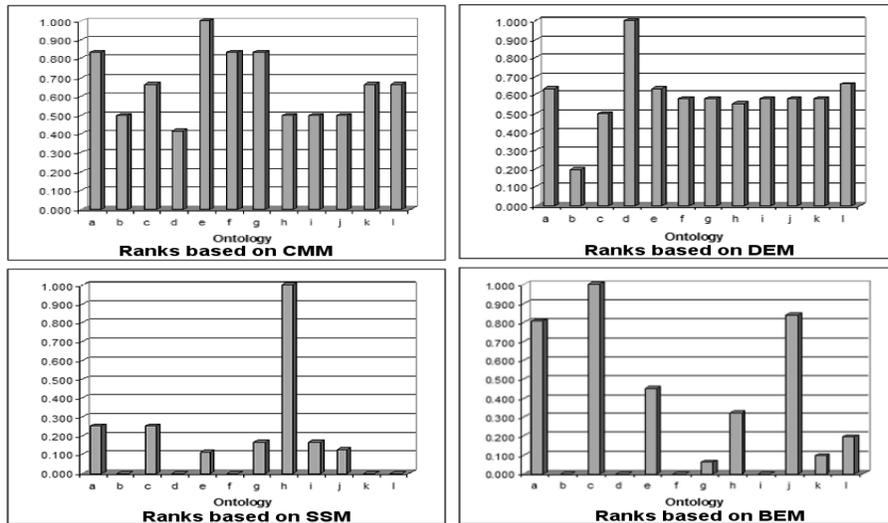


Fig. 5. Ontology rankings based on each measure separately

Ranking based on each measure separately is displayed in figure 5. When considered separately, none of the measures seemed to provide the same ranking list as when the measures were combined as will be discussed in the following section.

## 5 Evaluation

In order to evaluate the utility of the output of AKTiveRank, it is important to compare the results with those produced by some expert users. We have already conducted a small user-based experiment and used it to evaluate earlier versions of AKTiveRank [2]. Due to time constraints, we will use the results of that experiment again to evaluate our current results.

The users in our user-based experiment were presented with a general scenario, a set of screen shots of the relevant ontologies and a set of simple questions. Users were asked to rank the ontologies from the set presented, and were also given the opportunity to give comments and feedback. The total population sample was only four participants so we cannot make claims of any statistical accuracy or significance. Further and wider user-based experiments are planned for the very near future. The ranking as given by the users are listed in table 3:

Table 3. Ranks given by users

Ontology	Rank	Ontology	Rank	Ontology	Rank
a	2.5	b	12	c	11
d	9	e	2.5	f	5.5
g	5.5	h	1	i	10
j	7.5	k	4	l	7.5

When comparing the ranks produced by AKTiveRank in our experiments with the ranks generated from our user-based evaluation using the Pearson Correlation Coefficient (PCC), we get the value of 0.952. This value shows that the ranks produced by

AKTiveRank are very close to the ranks produced by our users (a value of 0 indicates no relation, and 1 indicates an exact linear relationship between the two datasets). Note that PCC value for Swoogle ranks against our user based results is -0.144, which indicates a very low and inversed correlation. Table 4 provides the PCC results above, as well as the PCC values when comparing each of our measure with the user results separately. It shows that the performance of each measure on its own was less than when they were combined (much higher PCC value when combined).

**Table 4.** Pearson Correlation Coefficient for each measures separately against rankings provided by users

Measure	Value
CMM	0.499
DEM	0.270
SSM	0.292
BEM	0.298
AktiveRank	0.952
Swoogle	-0.144

As can be seen in table 4, the measure that performed best when compared to the user results in table 3 was CMM, followed by BEM, SSM, then DEM. Based on this observation, the weights given to each measure when calculating the total score in our experiment were 0.4, 0.3, 0.2, and 0.1 respectively, to reflect the performance of each individual measures. These results are only representative of this experiment, and can not be generalised without further studies.

## 6 Conclusions and Future Work

In this paper we presented an extension to our previous work on ontology ranking based on an internal analysis of the concepts in the ontologies. The evaluation was based on four measures, but of course others may be added in the future if necessary. Even though our initial results are promising, a great deal of research and a much wider experiment are required before making any conclusive remarks about AKTiveRank’s measures.

The work presented here on the ranking of ontologies has been partly motivated by an awareness that ontologies are not artefacts like any other document on the web. They are crafted usually with considerable care where (for example) the importation of other ontologies usually has a certain significance. On the other hand, it is usual when constructing a domain specific ontology to import general ontologies like *foaf* which contain relatively little domain specific content. It is important to distinguish the function of an ontology from that of a web page. A web page is read by a human being and any links it may have may or may not be followed by the reader. In contrast, an ontology is designed to be read by a machine and any links it may have are by definition imports pointing to other ontologies which must be included. This poses a dilemma in ranking an ontology as to whether to include all imports or not. Because the imports tend to be high level general ontologies, they are relatively vacuous if the user is seeking a domain specific ontology. Further more if ontology  $O_1$  is dependent on ontology  $O_2$  to represent class  $c$ , then  $O_2$  will be evaluated separately anyway assuming it is included in the set retrieved.

It is very difficult to pinpoint the right selection of parameters or structural properties to investigate when ranking ontologies. The selection can be dependent on per-

sonal preference as well as use requirements (i.e. the purpose for which the ontology is intended). One focus of our future research will be to extend the user evaluation to include a larger number of human participants and a significant number of queries. Queries need to be posed to the system over a sufficient range of topics so as to allow confidence in the ranking methods we have used. Previous experience has shown it is difficult to present ontologies effectively to evaluators. Screen shots often show only a partial picture of the whole ontology, and some individuals prefer to examine the native OWL in understanding the ontology and making judgements. This is highly dependent on the background and skills of the user. Users must be given the freedom to browse the ontologies in an ontology editing tool such as Protégé[16] or Swoop [13], rather than given screen dumps or schema descriptions. For this reason, extensive user-based experiments are required to at least find out what are the properties that users tend to look at when judging the general quality or suitability of an ontology. Unlike ordinary search engines, where the user can be safely assumed to be relatively naive, with ontologies the typical user is either a knowledge engineer or software developer who has preconceptions of a technical nature.

Another area of future research lies in understanding further how these non-naive users set about finding an ontology so as to better model the user behaviour and tailor the system to that behaviour. In this regard, we have observed how users ask for ontologies in the Protégé mailing list and found that they tend to ask for topics (e.g. Transport, Algebra), which may not necessarily map to specific class names, but should rather be regarded as a general description of the domain. As consequence, we are currently investigating evaluating ontologies on their content coverage of a corpus [4], which is collected using the given general topic name (e.g. Cancer, Education)[12].

Other parameters can be taken into account, such as whether a class is defined or primitive (currently indirectly covered by the Density measure), or if the classes of interest are hubs or authoritative in a graph-network sense, which might increase their ontology's ranking.

The most appropriate criteria for searching for ontologies are still unclear. Swoogle is mainly based on keyword search, but other searching techniques can be imagined, based for example on the structure of ontologies or based on whether the ontologies meet certain requirements [14]. However, whatever the search mechanism is, there will always be a need for ranking. The ranking criteria will obviously have to be designed to fit the chosen search technique.

## Acknowledgments

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. Christopher Brewster has also been supported by the UK EPSRC under grant number GR/T22902/01

## References

1. H. Alani. Ontology construction from online ontologies. In *Proc. 15th International World Wide Web Conference*, Edinburgh, 2006.

2. H. Alani and C. Brewster. Ontology ranking based on the analysis of concept structures. In *3rd Int. Conf. Knowledge Capture (K-Cap)*, pages 51–58, Banff, Canada, 2005.
3. R. Angles and C. Gutierrez. Querying rdf data from a graph database perspective. In *Proc. of 2nd European Semantic Web conference, ESWC*, pages 346–360, Crete, 2005.
4. C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. Data driven ontology evaluation. In *Int. Conf. on Language Resources and Evaluation*, Lisbon, Portugal, 2004.
5. P. R. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Information Processing & Management*, 23(4):255–268, 1987.
6. L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A semantic web search and metadata engine. In *Proc. 13th ACM Conf. on Information and Knowledge Management*, Nov. 2004.
7. L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *Proc. 4th Int. Semantic Web Conf. (ISWC)*, pages 156–170, Galway, Ireland, 2005.
8. L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.
9. A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. A theoretical framework for ontology evaluation and validation. In *Semantic Web Applications and Perspectives (SWAP) – 2nd Italian Semantic Web Workshop*, Trento, Italy, 2005.
10. A. Gomez-Perez. Some ideas and examples to evaluate ontologies. In *11th Conference on Artificial Intelligence for Applications*. IEEE Computer Society, 1995.
11. N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65, 2002.
12. M. Jones and H. Alani. Content-based ontology ranking. In *Proceedings of the 9th Int. Protege Conf.*, Stanford, CA, 2006.
13. A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, and J. Hendler. Swoop: A 'web' ontology editing browser. *Journal of Web Semantics*, 4(2), 2005.
14. A. Lozano-Tello and A. Gomez-Perez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2), 2005.
15. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proc. European Conf. on Knowledge Acquisition and Management (EKAW)*, pages 251–263, Madrid, 2002.
16. N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*, pages 60–71, 2001.
17. C. Patel, K. Supekar, Y. Lee, and E. Park. 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.
18. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Trans. on Systems Management and Cybernetics*, 19(1):17–30, 1989.
19. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
20. E. Rosch. *Principles of Categorization*. In E. Rosch and B. B. Lloyd editors. Cognition and Categorization, Lawrence Erlbaum, Hillsdale, New Jersey, 1978.
21. G. Spanoudakis and P. Constantopoulos. Similarity for analogical software reuse: A computational model. In *Proc. 11th European Conf. on AI, ECAI'94*, pages 18–22, 1994.
22. K. Supekar. A peer-review approach for ontology evaluation. In *8th Int. Protege Conf.*, pages 77–79, Madrid, Spain, July 2005.
23. A. Tversky. Features of similarity. *Psychological Review*, 84(4), 1977.
24. P. C. Weinstein and W. P. Birmingham. Comparing concepts in differentiated ontologies. In *Proc. 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW' 99)*, Banff, Alberta, Canada, 1999.