

# Metrics for Ranking Ontologies

Harith Alani  
Intelligence, Agents, Multimedia Group  
School of Electronics and Computer Science  
University of Southampton  
Southampton, UK  
h.alani@ecs.soton.ac.uk

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

## ABSTRACT

Representing knowledge using domain ontologies has shown to be a useful mechanism and format for managing and exchanging information. Due to the difficulty and cost of building ontologies, a number of ontology libraries and search engines are coming to existence to facilitate reusing such knowledge structures. The need for ontology ranking techniques is becoming crucial as the number of ontologies available for reuse is continuing to grow. In this paper we present AKTiveRank, a prototype system for ranking ontologies based on the analysis of their structures. We describe the metrics used in the ranking system and present an experiment on ranking ontologies returned by a popular search engine for an example query.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Selection process

## Keywords

Ontology ranking, ranking metrics

## 1. INTRODUCTION

Ontologies are the backbone of knowledge representation for the Semantic Web. The challenges involved in building ontologies in terms of time, effort, and domain specific knowledge have meant that large scale use of ontologies encounters the classic “knowledge acquisition bottleneck”. In order to minimise this, one of the major advantages claimed of ontologies is the potential for the “reuse” of knowledge. Publicly available ontologies are to be reused, modified, extended, and pruned as required, thereby avoiding the huge effort of starting from scratch.

A number of ontology libraries currently exist, hosting various ontology files. Examples of such libraries include Ontolingua<sup>1</sup>, the DAML library<sup>2</sup>, the Protège OWL library<sup>3</sup>, etc. The ontology search facilities provided by these libraries are at best limited to term search, making it difficult for the user to select the relevant ontologies. As the number of publicly available ontologies increases, this problem is bound to

<sup>1</sup><http://www-ksl-svc.stanford.edu:5915/>

<sup>2</sup><http://www.daml.org/ontologies/>

<sup>3</sup><http://protege.stanford.edu/plugins/owl/owl-library/>

get worse. There is a need for more and more ontologies to be constructed and made available. However, as this occurs, so the reuse of this knowledge becomes an ever greater problem.

In order to achieve an effective level of knowledge reuse, we need search engines capable of helping us find the ontologies we are looking for. Some ontology search engines have been developed that can provide lists of ontologies that contain specific search terms, such as Swoogle [5] and OntoSearch [23]. Such search engines are a good step forward, but more is required in terms of ontology search if reuse is to become a reality [1]. 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 the user 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 knowledge engineer.

This paper presents a prototype of an ontology ranking system which applies a number of analytic methods to rate each ontology based on how well it represents the given search terms. Related work concerning ontology searching 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. Conclusions and future work are discussed in the final section of the paper.

## 2. RELATED WORK

Lack of automatic, well grounded, methodologies to evaluate ontologies may seriously hinder their adoption by the industry and the wider web community [8]. Ontologies may be assessed from different angles, such as how the ontologies have been rated and reviewed by users (e.g. [20]), how well they meet the requirements of certain evaluation tests (e.g. [10]) or general ontological properties (e.g. [12]).

Gangemi and colleagues [8] 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. [9]). Structural evaluation focuses on the structural properties of the ontology as a graph (e.g. [2]).

This paper studies the evaluation of ontologies based on a number of internal structural measures. The purpose of the evaluation is to enable a system to rank ontologies returned by search engines according to how well the ontologies perform under certain measures.

Ranking has always been at the heart of information retrieval. This became even more apparent given the enormous size of the web and its continuous expansion. Google uses the PageRank [15] method to rank documents based on hyperlink analysis. Some ontology search engines adopted a PageRank-like method to rank ontologies by analyses links and referrals between the ontologies in the hope of identifying the most popular ones (e.g. Swoogle [5, 6] and OntoKhoj [16]). 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 ontologies at all [5]. Poor connectivity would certainly produce poor PageRank results.

Furthermore, a popular ontology does not necessarily indicates a good representation of all the concepts it covers. Popularity does not necessarily correlate with ‘good’ or appropriate representations of knowledge. For example, supposing an engineer was looking for an ontology about “students”, there could be an ontology about the 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. That ontology might have become popular due to its coverage of publications and research topics, rather than for it’s coverage of student related concepts.

Another challenge for ranking ontologies is searching for multiple terms. For example, if searching for “pet” AND “food”, then an ontology that has such classes in good structural proximity to each other is better than one where those classes are further apart. Various formulae exist in the literature to measure similarities of terms within semantic networks, and these can be used equally to measure structural proximity.

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 (no. 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.

When a query is received, AKTiveRank queries Swoogle (no. 3) for the given search terms and retrieves the ontology URIs from the results page returned by Swoogle. Swoogle only allows for keyword-based queries to be made when searching for ontologies. AKTiveRank does not search for ontologies, but only attempts to rank them. Currently, AKTiveRank relies entirely on Swoogle for searching ontologies, and therefore queries submitted to AKTiveRank are meant to resemble Swoogle queries by being based on keywords.

Once a list of ontology candidates is gathered from Swoogle, AKTiveRank starts to check whether those ontologies are already stored in a Jena MySQL database back-end (no. 4), and if not, download 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. Some of the analysis of ontology structures that AKTiveRank performs for the ranking is also undertaken using Jena’s API.

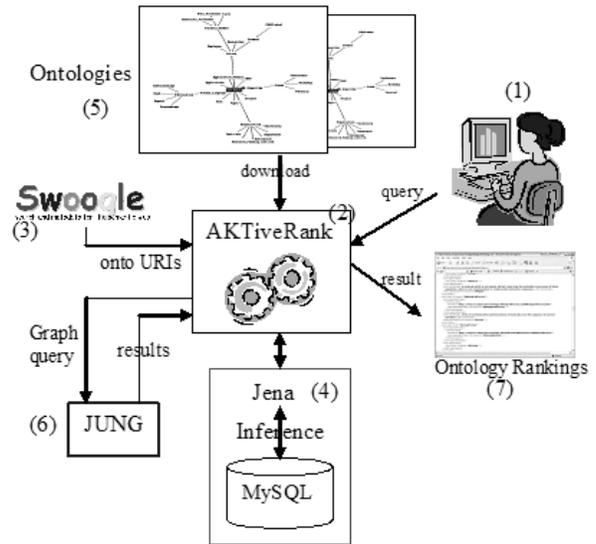


Figure 1: AKTiveRank Architecture

Existing RDF query languages are not well suited for graph queries [3]. 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 types of assessments (measures) for each ontology to measure the rankings. 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.

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 more central a class is in the hierarchy, the more likely it is for it to be well analysed and fully represented [19]. However, in some experiments we found a few ontologies that placed our concept of interest as a near-top-level concept. 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 renders CEM somewhat redundant. The Density measure calculates the information content of a class. This observation backs Rosch’s studies [19] which showed that classes at mid-hierarchical levels tend to have more detail than others.

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.

### 3.1.1 Class Match Measure

The Class Match Measure (CMM) is meant to evaluate the coverage of an ontology for the given search terms. AK-TiveRank 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 contains 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”.

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 irrespectively of whether an exact match has been found or not. 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.

**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.

### 3.1.2 Density Measure

When searching for a “good” representation of a specific concept, one would expect to find a certain degree of detail in the representation of the knowledge concerning that concept. This may include how well the concept is further specified (the number of subclasses), the number of attributes 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 numbers of 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  $S = \{S_1, S_2, S_3, S_4\} = \{relations[c], superclasses[c], subclasses[c], siblings[c]\}$

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

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

where  $w_i$  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  $o$ .

### 3.1.3 Semantic Similarity Measure

Similarity measures have often been used in information retrieval systems to provide better ranking for query results. 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 [18]. The measure he used is based on the comparison of shared features, which was first proposed in [21]. Another common-feature based similarity is the shortest-path measure, introduced by Rada [17]. He argues that the more relationships objects have in common, the closer they will be in an ontology. Rada used this measure to help rank biomedical documents which were represented in a semantic knowledge-base. Variations of these techniques have been used to measure similarity between whole ontology structures [13, 22].

The Semantic Similarity Measure (SSM) calculates how close the classes that matches the search terms are in an ontology. The motivation for this is that ontologies which position concepts further away from each other are less likely to represent the knowledge in a coherent and compact manner<sup>4</sup>. The SSM formula used here is based on the shortest path measure defined in [17]. SSM is measured from the minimum number of links that connects a pair of concepts. These links can be isA relationships or 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  $c_j$

$$ssm(c_i, c_j) = \begin{cases} \frac{1}{length(\min_{p \in P} \{c_i \xrightarrow{p} c_j\})} & : \text{if } i \neq j \\ 0 & : \text{if } i = j \end{cases} \quad (8)$$

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

where  $n$  is the number of matched classes.

### 3.1.4 Betweenness Measure

<sup>4</sup>Further studies are required to find whether or not this assumption is dependent on certain ontology properties, such as size or level of detail

One of the algorithms that JUNG provides is Betweenness [7]. 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 is that if a class has a high betweenness value in an ontology then this class is *central* to that ontology.

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

**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  $c$ .

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

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  $o$ .

### 3.1.5 Total 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.

**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. 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.

## 4. EXPERIMENT

In this section we report the results of running AKTiveRank over an example query submitted to Swoogle<sup>5</sup>.

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. Further experiments will be required to identify the best mix of weights to reach optimum ranking results. The relative weights for these measures

<sup>5</sup>Note that we used Soogle 2005 in our experiment. The results of the search may thus be somewhat different if using the latest version of Swoogle which became available near the completion of our experiment

are selected based on how well each measure performed in our evaluation as will be discussed in section 5.

Now lets assume that we need to find an OWL ontology that represents the concepts of “University” and “Student”. 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 the ontologies returned by Swoogle 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 save space. It is worth mentioning that Swoogle returned those duplicated ontologies in very different orders. For example the *exact* “koala” ontology was returned under 3 URLs in the 2nd<sup>6</sup>, 9th<sup>7</sup>, and 18th<sup>8</sup> position.

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 resulting list shown in table 1, it produced the values given in table 2, which are displayed in figure 2.

**Table 2: AKTiveRank results**

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

## 4.1 Analysis of Results

From the results of this experiment, it can be seen that ontology *a* scored the highest value in AKTiveRank. The ontologies *c* and *h* where 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 places last in the ranked list. Even though this ontology contains classes labelled “Student” and “University”, but those classes are not 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

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

<sup>7</sup><http://www.ling.helsinki.fi/kit/2004k/ct1310semw/Protege/koala.owl>

<sup>8</sup><http://www.iro.umontreal.ca/~touzanim/Ontology/koala.owl>

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

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

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

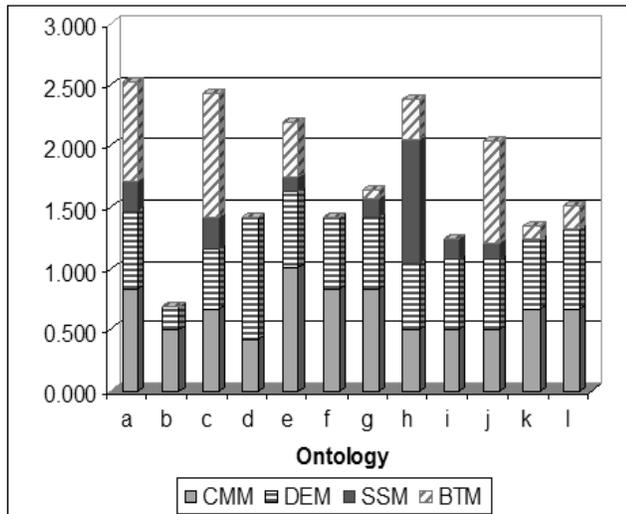


Figure 2: student + university.

paths that cross via the imaginary *owl:Thing* class are ignored.

The ontology that scored the highest in the Class Match measure (CMM, section 3.1.1) 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.

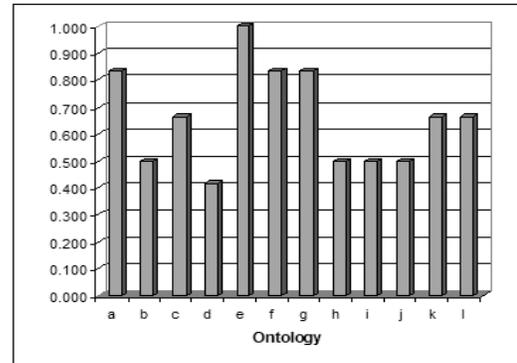


Figure 3: Ranks based on CMM.

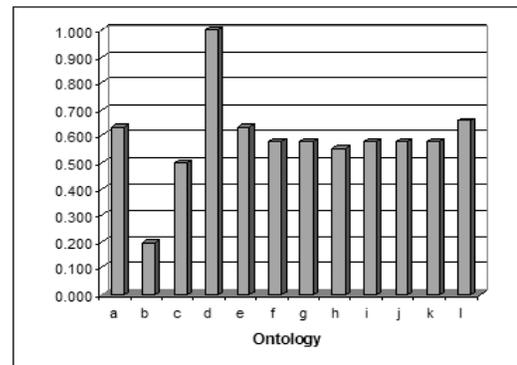


Figure 4: Ranks based on DEM

Ranking based on each measure separately is displayed in figures 3, 4, 5 and 6. When taken separately, none of the measures provided the same ranking list as when the measures were combined as will be discussed in the following section.

## 5. EVALUATION

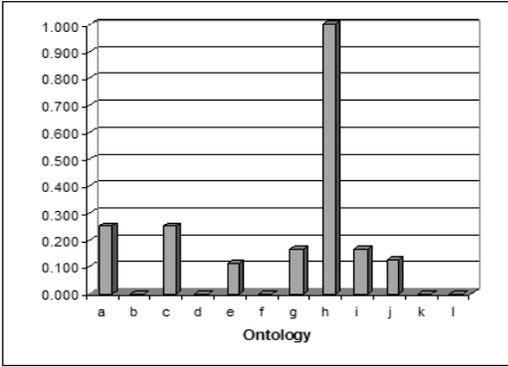


Figure 5: Ranks based on SSM

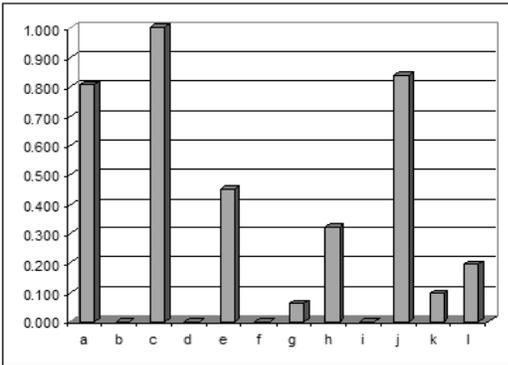


Figure 6: Ranks based on BEM

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:

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

Table 3: Ranks given by users

Ontology	User Given 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

Table 4: Pearson Correlation Coefficient for each measures separately against expert rankings for our experiment

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

the performance of each measure on its own was less than when they were combined (much higher PCC value when combined).

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.

## 6. CONCLUSIONS

In this paper we have presented a specific approach to constructing an ontology ranking engine based on an internal analysis of the concepts in the ontologies. Even though our initial results are promising, but a much wider experiment is required before making any conclusive remarks about AKTiveRank’s measures.

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 personal preference as well use requirements (i.e. purpose for using the ontology). 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. Users must be given the freedom to browse the ontologies in an ontology editing tool such as Protégè[14] or Swoop [11], rather than given screen dumps or schema descriptions.

Another important point to raise is the criteria for searching for ontologies. Swoogle is 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 [12]. 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.

## 7. 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. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of the EPSRC or any other member of the AKT IRC. Special thanks to Dr Srinandan Dasmahapatra for his valuable help, to Dr. Daniela Petreli for help with the questionnaire, and to the four scientists who participated in our evaluation. Thanks also to Ben O'Neil for building the JUNG interface.

## 8. 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] 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.
- [6] 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.
- [7] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.
- [8] 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.
- [9] A. Gomez-Perez. Some ideas and examples to evaluate ontologies. In *11th Conference on Artificial Intelligence for Applications*. IEEE Computer Society, 1995.
- [10] N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65, 2002.
- [11] 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.
- [12] A. Lozano-Tello and A. Gomez-Perez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2), 2005.
- [13] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW)*, pages 251–263, Madrid, Spain, 2002. Springer.
- [14] 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, March/April 2001.
- [15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.
- [16] 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.
- [17] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.
- [18] 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.
- [19] E. Rosch. *Principles of Categorization*. in E. Rosch and B. B. Lloyd editors. *Cognition and Categorization*, Lawrence Erlbaum, Hillsdale, New Jersey, 1978.
- [20] K. Supekar. A peer-review approach for ontology evaluation. In *8th Int. Protege Conf.*, pages 77–79, Madrid, Spain, July 2005.
- [21] A. Tversky. Features of similarity. *Psychological Review*, 84(4), 1977.
- [22] 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.
- [23] Y. Zhang, W. Vasconcelos, and D. Sleeman. Ontosearch: An ontology search engine. In *Proc. 24th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, UK, 2004.