Ontology Search: An Empirical Evaluation

Anila Sahar Butt[1,2], Armin Haller[1], and Lexing Xie[2]

[1] CSIRO CCI, firstname.lastname@csiro.au
[2] Australian National University, firstname.lastname@anu.edu.au

Abstract. Much of the recent work in Semantic Search is concerned with addressing the challenge of finding entities in the growing Web of Data. However, alongside this growth, there is a significant increase in the availability of ontologies that can be used to describe these entities. Whereas several methods have been proposed in Semantic Search to rank entities based on a keyword query, little work has been published on search and ranking of resources in ontologies. To the best of our knowledge, this work is the first to propose a benchmark suite for ontology search. The benchmark suite, named CBRBench[3], includes a collection of ontologies that was retrieved by crawling a seed set of ontology URIs derived from prefix.cc and a set of queries derived from a real query log from the Linked Open Vocabularies search engine. Further, it includes the results for the ideal ranking of the concepts in the ontology collection for the identified set of query terms which was established based on the opinions of ten ontology engineering experts.
We compared this ideal ranking with the top-k results retrieved by eight state-of-the-art ranking algorithms that we have implemented and calculated the precision at k, the mean average precision and the discounted cumulative gain to determine the best performing ranking model. Our study shows that content-based ranking models outperform graph-based ranking models for most queries on the task of ranking concepts in ontologies. However, as the performance of the ranking models on ontologies is still far inferior to the performance of state-of-the-art algorithms on the ranking of documents based on a keyword query, we put forward four recommendations that we believe can significantly improve the accuracy of these ranking models when searching for resources in ontologies.

1 Introduction

The growth of Linked Data in recent years has given rise to the need to represent knowledge based on ontologies. Prefix.cc[4], a service to register prefixes, counts about ~1250 such ontologies (April 2014) whereby many cover similar domains, e.g. our crawl found the concept *Person* to exist in 379 ontologies. One of the major advantages claimed of ontologies, however, is the potential of "reuse" opposed to creating a new ontology from scratch. Consequently, finding the right ontology, or more specifically classes and properties within ontologies that match the intended meaning for a specific use case is an important task that is becoming increasingly difficult.

The Linked Open Vocabularies (LOV) search engine[5], initiated in March 2011, is to the best of our knowledge, the only purpose-built ontology search engine available on the Web with an up-to-date index. It uses a ranking algorithm based on the term popularity in Linked Open Data (LOD) and in the LOV ecosystem [22].

[3] https://zenodo.org/record/11121
[4] http://prefix.cc
[5] http://lov.okfn.org

There are also some ontology libraries available that facilitate the locating and retrieving of potentially relevant ontology resources [13]. Some of these libraries are domain-specific such as the Open Biological and Biomedical Ontologies library[6] or the BioPortal [14], whereas others are more general such as OntoSearch [20] or the TONES Ontology Repository[7]. However, as discussed by Noy & d'Aquin [13] only few libraries support a keyword search and basic ranking, and only one (Cupboard [2]) supports a ranking of ontologies based on a user query using an information retrieval algorithm (i.e. tf-idf), while no library supports the ranking of resources within the registered ontologies.

Semantic Search engines such as Swoogle [4] (which was initially developed to rank ontologies only), Sindice.com [21], Watson [3], or Yars2 [8] do allow a search of ontology resources through a user query. The ranking in these search engines follows traditional link-based ranking methods [10], in particular adapted versions of the PageRank algorithm [15], where links from one source of information to another are regarded as a 'positive vote' from the former to the latter. Often, these ranking schemes also take the provenance graph of the data into account [9]. Other strategies, mainly based on methods proposed in the information retrieval community, are employed in Semantic Search [5], but what all these methods have in common is that they are targeted to rank entities, but do not work well for ranking classes and properties in ontologies [4, 1].

The task of ranking resources defined in ontologies can be based on many different criteria [6], for example, how well an ontology meets the requirements of certain evaluation tests (e.g. [7]) or on methods to evaluate general properties of an ontology based on some requirement (e.g. [11]). However, only limited work has been proposed to rank the returned resources based on a user posed keyword query such that the most relevant results appear higher in the list. Alani et al. [1] propose four measures (i.e. Semantic Similarity, Betweenness, Density and Class Match Measure) to evaluate different representational aspects of the ontology and calculate its ranking.

In the information retrieval community many algorithms, such as the vector space model, the boolean model, BM25, tf-idf, etc. have been proposed to identify and rank a small number of potentially relevant documents through a top-k document retrieval. To the best of our knowledge, no systematic study has been conducted to compare the performance of these state-of-the-art ranking techniques on the task of ranking resources in ontologies. For our study we have implemented eight ranking algorithms, four of which have been proposed by the information retrieval community whereas the others were adapted for the ranking of ontologies by Alani et al [1]. We defined a set of queries derived from a real query log, and computed the ranking for these queries on a collection of ontology resources that we have crawled with a seed set of ontology URIs derived from prefix.cc. We computed a baseline ranking and established a ground truth by asking ten ontology engineers to manually rank ontologies based on a given search term from the collection of resources obtained by the baseline ranking. We compared the ground truth derived through the human evaluation with the results from each of the ranking algorithms. We calculated the precision at k, the mean average precision and the discounted cumulative gain of the ranking algorithms in comparison to a ground truth to determine the best model for the task of ranking resources/ontologies. The contribution of this paper are:

[6] `http://www.obofoundry.org/`
[7] `http://owl.cs.manchester.ac.uk/repository/`

– a design of a benchmark suite named CBRBench, for Canberra Ontology Ranking Benchmark, including an ontology collection, a set of queries and a ground truth established by human experts for evaluating ontology ranking algorithms,
– a methodology for resource ranking evaluation where we discuss many of the decision that need to be made when designing a search evaluation framework for resources defined in ontologies,
– the evaluation of eight ranking algorithms through these benchmarks, and
– a set of recommendations derived from an analysis of our experiment that we believe can significantly improve the performance of the ranking models.

The remainder of this paper is organised as follows. We begin with a discussion of the ranking algorithms that we have implemented for this experiment in Section 2. In Section 3, we describe the evaluation setup. We then present the results and a result analysis in Section 4. Section 5 discusses some recommendations for the improvement of the ranking models for ontology search, before we conclude in Section 6.

2 Ranking algorithms

We have chosen eight different ranking models that are commonly used for ranking documents and/or ontologies and applied them on the task of ranking resources/ontologies according to their relevance to a query term. These eight ranking models can be grouped in two different categories.

1. **Content-based Ranking Models**: tf-idf, BM25, Vector Space Model and Class Match Measure.
2. **Graph-based Ranking Models**: PageRank, Density Measure, Semantic Similarity Measure and Betweenness Measure.

Because of the inherit graph structure of ontologies, graph-based ranking models can be used for ranking as such. However, content-based ranking models (e.g. tf-idf, BM25 and Vector Space Model) need to be tailored towards ontologies so that instead of using a word as the basic unit for measuring, we are considering a resource r in an ontology as the measuring unit. Therefore, the relevance of a query word to the ontology is the sum of the relevance of all the resources that match the query term. For tf-idf we compute the relevance score of the resource, all other algorithms generate a cumulative relevance score for the ontology and resources are ranked according to the relevance score of their corresponding ontology. The matched resource set for each term/word is selected from a corpus if a word exists in the value of the 1) `rdfs:label` 2) `rdfs:comment`, or 3) `rdfs:description` property of that resource or if the word is part of the URI of the resource. As most of the existing adaptations of graph ranking models for ontology ranking do not compute a ranking for properties in an ontology we only consider the ranking of classes/concepts in this study. However, it turns out that only 2.6% of all resources in our corpus (cf. Section 3) are properties.

In the following sections we introduce all ranking models, and describe the choices we made to adapt them for the ranking of resources in ontologies. Common notations used in the following sections are shown in Table 2.

2.1 tf-idf

Term frequency inverse document frequency (tf-idf) [18] is an information retrieval statistic that reflects the importance of a word to a document in a collection or

Variable	Description
$\mathbb{O}$	Corpus: The ontology repository
N	Number of ontologies in $\mathbb{O}$
O	An ontology: $O \in \mathbb{O}$
r	A resource uri: $r \in O$ & $r \in URI$
z	Number of resources in O
Q	Query String
q_i	query term i of Q
n	number of keywords in Q
σ_O	set of matched uris for Q in O
$\sigma_O(q_i)$	set of matched uris for q_i in O : $\forall$ $r_i \in \sigma_O$, $r_i \in O$ & r_i matches q_i
m	number of uris in $\sigma_O(q_i)$

Table 1. Notation used

corpus. For ranking ontologies we compute the importance of each resource r to an ontology O in a ontology repository, where $r \in R : R =$ URI only (i.e. excluding blank nodes and literals).

$$\begin{aligned} tf(r,O) &= 0.5 + \frac{0.5 * f(r,O)}{max\{f(r_i,O) : r_i \in O\}} \\ idf(r,\mathbb{O}) &= log\frac{N}{|\{O \in \mathbb{O} : r \in O\}|} \\ tf - idf(r,O,\mathbb{O}) &= tf(r,O) * idf(r,O) \end{aligned} \tag{1}$$

Here *tf(r,O)* is the term frequency for resource r in *O*. *tf(r,O)* is the frequency of r (number of times r appears in O) divided by the maximum frequency of any resource r_i in O. The inverse document frequency $idf(r,\mathbb{O})$ is a measure of commonality of a resource across the corpus. It is obtained by dividing the total number of ontologies in the corpus by the number of documents containing the resource r, and then computing the logarithm of that quotient. The final score of r for this query is the tf-idf value of r.

$$Score(r,Q) = tf - idf(r,O,\mathbb{O}) : \forall r\{\exists q_i \in Q : r \in \sigma(q_i)\} \tag{2}$$

2.2 BM25

BM25 [17] is a ranking function for document retrieval used to rank matching documents according to their relevance to a given search query. Given a query Q, containing keywords q_1, ..., q_n, the BM25 score of a document d is computed by:

$$score(d,Q) = \sum_{i=1}^{n} idf(q_i,d)\frac{tf(q_i,d) * k + 1}{tf(q_i,d) + k * (1 - b + b * (\frac{|d|}{avgdl}))} \tag{3}$$

where $tf(q_i,d)$ is the term frequency and $idf(q_i,d)$ is the inverse document frequency of the word q_i. |d| is the length of the document d in words, and $avgdl$ is the average document length in the text collection from which the documents are drawn. k_1 and b are free parameters, usually chosen, in absence of an advanced optimisation, as $k_1 \in [1.2,2.0]$ and $b = 0.75$.

In order to tailor this statistic for ontology ranking we compute the sum of the score of each $r_j \in \sigma_O(q_i)$ for each query term q_i rather than computing the score for q_i. For the current implementation we used $k_1 = 2.0$, $b = 0.75$ and |O| = total number of terms (i.e. 3 * |axioms|) in the ontology. The final score of the ontology is computed as:

$$score(O,Q) = \sum_{i=1}^{n} \sum_{\forall r_j : r_j \in \sigma_O(q_i)} idf(r_j, O) \frac{tf(r_j, O) * k + 1}{tf(r_j, O) + k * (1 - b + b * (\frac{|O|}{avgol}))} \quad (4)$$

2.3 Vector Space Model

The vector space model [19] is based on the assumptions of the document similarities theory where the query and documents are represented as the same kind of vector. The ranking of a document to a query is calculated by comparing the deviation of angles between each document vector and the original query vector. Thus, the similarity of a document to a query is computed as under:

$$sim(d,Q) = \frac{\sum_{i=1}^{n} w(q_i, d) * w(q_i, Q)}{|d| * |Q|} \quad (5)$$

Here $w(q_i, d)$ and $w(q_i, Q)$ are weights of q_i in document d and query Q respectively. $|d|$ is the document norm and $|Q|$ is the query norm. For this implementation, we are considering the *tf-idf* values of a query term as weights. Therefore, the similarity of an ontology to query Q is computed as:

$$\begin{aligned}
sim(O,Q) &= \frac{\sum_{i=1}^{n} tf - Idf(q_i, O) * tf - idf(q_i, Q)}{|O| * |Q|} \\
tf - idf(q_i, O) &= \sum_{j=1}^{m} tf - idf(r_j, O) : r_j \in \sigma_O(q_i) \\
tf - idf(q_i, Q) &= \frac{f(q_i, Q)}{max\{f(q, Q) : q \in Q\}} * log \frac{N}{|\{O \in \mathbb{O} : r \in O \& r \in \sigma_O(q_i)\}|} \\
|O| &= \sqrt{\sum_{i=1}^{z} (tf - idf(r_i, O))^2} \\
|Q| &= \sqrt{\sum_{i=1}^{n} (tf - idf(q_i, O))^2}
\end{aligned} \quad (6)$$

2.4 Class Match Measure

The Class Match Measure (CMM) [1] evaluates the coverage of an ontology for the given search terms. It looks for classes in each ontology that have matching URIs for 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 score higher than others, and exact matches are regarded as better than partial matches. The score for an ontology is computed as:

$$score_{_{CMM}}(O,Q) = \alpha score_{_{EMM}}(O,Q) + \beta score_{_{PMM}}(O,Q) \quad (7)$$

where $score_{_{CMM}}(O,Q)$, $score_{_{EMM}}(O,Q)$ and $score_{_{PMM}}(O,Q)$ are the scores for class match measure, exact match measure and partial match measure for the ontology O with respect to query Q, α and β are the exact matching and partial matching weight factors respectively. As exact matching is favoured over partial matching, therefore $\alpha > \beta$. For our experiments, we set $\alpha = 0.6$ and $\beta = 0.4$ (as proposed in the original paper [1]).

$$score_{_{EMM}(O,Q)} = \sum_{i=1}^{n}\sum_{j=1}^{m} \varphi(r_j, q_i) : r_j \in \sigma_O(q_i)$$
$$\varphi(r_j, q_i) = \begin{cases} 1 & \text{if label}(r_j) = q_i \\ 0 & \text{if label}(r_j) \neq q_i \end{cases} \tag{8}$$

$$score_{_{PMM}}(O,Q) = \sum_{i=1}^{n}\sum_{j=1}^{m} \psi(r_j, q_i) : r_j \in \sigma_O(q_i)$$
$$\psi(r_j, q_i) = \begin{cases} 1 & \text{if label}(r_j) \text{ contains } q_i \\ 0 & \text{if label}(r_j) \text{ does not contain } q_i \end{cases} \tag{9}$$

2.5 PageRank

PageRank [15] is a hyperlink based iterative computation method for document ranking which takes as input a graph consisting of nodes and edges (i.e. ontologies as nodes and `owl:imports` properties as links in this implementation). In each successive iteration the score of ontology o is determined as a summation of the PageRank score in the previous iteration of all the ontologies that link (imports) to ontology O divided by their number of outlinks (owl:imports properties). For the k_{th} iteration the rank of ontology O i.e. ($score_k(O)$) is given as under:

$$score_k(O) = \frac{\sum_{j \in deadlinks(\mathbb{O})} R_{k-1}(j)}{n} + \sum_{i \in ininks(O)} \frac{R_{k-1}(i)}{|outdegree(i)|}$$
$$score_k(O) = d * score_k(O) + \frac{1-d}{n} \tag{10}$$

Here $deadlinks(\mathbb{O})$ are ontologies in corpus $\mathbb{O}$ that have no outlinks, i.e. they never import any other ontology. All nodes are initialised with an equal score (i.e. $\frac{1}{n}$, where n is total number of ontologies in $\mathbb{O}$ before the first iteration. In the experimental evaluation, we set the damping factor d equal to 0.85 (common practise) and we introduced missing `owl:imports` link among ontologies based on reused resources.

2.6 Density Measure

Density Measure (DEM) [1] is intended to approximate the information content of classes and consequently the level of knowledge detail. This includes how well the concept is further specified (i.e. the number of subclasses), the number of properties associated with that concept, number of siblings, etc. Here $score_{_{DEM}}(O,Q)$ is the density measure of ontology O for query Q. $\Theta(r_j, q_i)$ is the density measure for resource r_j and w is a weight factor set for each dimensionality i.e. sub classes = 1,

super classes = 0.25, relations = 0.5 and siblings = 0.5 and $k = n * m$ (i.e. number of matched r) for query Q.

$$
\begin{aligned}
score_{_{DEM}}(O,Q) &= \frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{m}\Theta(r_j) : r_j \in \sigma_O(q_i) \\
\Theta(r_j) &= \sum_{s_k \in S} w_{s_k}|s_k| \\
S &= \{s_{sub}, s_{sup}, s_{sib}, s_{rel}\} \\
w &= \{1, 0.25, 0.5, 0.5\}
\end{aligned} \tag{11}
$$

2.7 Semantic Similarity Measure

The Semantic Similarity Measure (SSM) calculates how close the concepts of interest are laid out in the ontology structure. The idea is, if the concepts are positioned relatively far from each other, then it becomes unlikely for those concepts to be represented in a compact manner, rendering their extraction and reuse more difficult. $score_{_{SSM}}(O,Q)$ is the semantic similarity measure score of ontology O for a given query Q. It is a collective measure of the shortest path lengths for all classes that match the query string.

$$
\begin{aligned}
score_{_{SSM}}(O,Q) &= \frac{1}{z}\sum_{i=1}^{z-1}\sum_{j=i+1}^{z}\Psi(r_i,r_j) : \forall_{q\in Q}((r_i,r_j) \in \sigma_O)) \\
\Psi(r_i,r_j) &= \begin{cases} \frac{1}{length(min_{p\in P}\{r_i \xrightarrow{p} r_j\})} & \text{if i} \neq \text{j} \\ 1 & \text{if i} = \text{j} \end{cases} \\
z &= |(r_i,r_j)|
\end{aligned} \tag{12}
$$

2.8 Betweenness Measure

The Betweenness Measure (BM)[1] is a measure for a class on how many times it occurs on the shortest path between other classes. This measure is rooted on the assumption that if a class has a high betweenness value in an ontology then this class is graphically central to that ontology. The betweenness value of an ontology is the function of the betweenness value of each queried class in the given ontologies. The ontologies where those classes are more central receive a higher BM value.

$score_{_{BM}}(O,Q)$ is the average betweenness value for ontology O and k is the number of matched resources from O for Q. The betweenness measure for resource r_j i.e. $\vartheta(r_j,q_i)$ is computed as:

$$
\begin{aligned}
score_{_{BM}}(O,Q) &= \frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{m}\vartheta(r_j,q_i) : r_j \in \sigma_O(q_i) \\
\vartheta(r_j,q_i) &= \sum_{r_x \neq r_y \neq r_j} \frac{\lambda(r_x, r_y(r_j))}{\lambda(r_x,r_y)}
\end{aligned} \tag{13}
$$

where $\lambda(r_x,r_y)$ is the number of the shortest path from r_x and r_y and $\lambda(r_x,r_y(r_j))$ is the number of shortest paths from r_x and r_y that passes through r_j.

3 Experiment setup

To compare and evaluate the implemented ranking models we developed a benchmark suite named CBRBench, for Canberra Ontology Ranking Benchmark, which includes a collection of ontologies, a set of benchmark queries and a ground truth established by human experts. The CBRBench suite is available at `https://zenodo.org/deposit/9817/`.

3.1 Benchmark Ontology collection

To the best of our knowledge there exists no benchmark ontology collection for ranking of ontologies. To derive at a representative set of ontologies used on the Web, we used the namespaces registered at prefix.cc[8] as our set of seed ontology URIs. We crawled all registered prefix URIs and for each successfully retrieved ontology (we encountered hundreds of deadlinks and non-ontology namespaces) we also followed its import statements until no new ontologies were found. This resulted in 1022 ontologies that we used as our benchmark collection. In total these ontologies define more than 5.5M triples, including ~280k class definitions and ~7.5k property definitions. We stored each ontology seperately as a named graph in a Virtuoso database.

3.2 Benchmark query terms

To test the ranking algorithms on a representative set of query terms we have used the query log[9] of the Linked Open Vocabularies (LOV) search engine [22] as input. We ranked the most popular search terms in the log covering the period between 06/01/2012 and 16/04/2014 based on their popularity. For the most popular query terms we checked through a boolean search if there is a representative sample of relevant resources available in our benchmark ontology collection that at least partially match the query term. We included ten search terms in our corpus where there were at least ten relevant ontology classes in the result set. The chosen search terms and their popularity rank within the Linked Open Vocabularies search log are shown in Table 2. All queries are single word queries – that is for two reasons. First, only about 11% of all queries posed on the LOV search engine use compound search queries and no compound query was among the 200 most used queries and second, for no compound query in the top 1000 query terms did the benchmark collection contain enough relevant resources to derive at a meaningful ranking.

Although shallow evaluation schemes are preferred in web search engine evaluations [16] we opted for a deep evaluation scheme for two reasons. First, there is only a limited set of knowledge domains where there is a sufficient number of ontologies available on the Web, and second, for the domains with a sufficient number of ontologies, many ontologies exist that define or refine similar concepts. This assumption is confirmed by the high number of matching classes for the terms in our query set (see for example Table 3).

3.3 Establishing the ground truth

We conducted a user study with ten human experts who were sourced from the Australian National University, Monash University, the University of Queensland and the CSIRO. Eight of the evaluators considered themselves to possess "Expert

[8] `http://www.prefix.cc`

[9] See `http://lov.okfn.org/dataset/lov/stats/searchLog.csv`

Table 2. Query terms

Search Term	Rank
person	1
name	2
event	3
title	5
location	7
address	8
music	10
organization	15
author	16
time	17

Table 3. Ranking of "Person" in ground truth

URI	Rank
http://xmlns.com/foaf/0.1/Person	1
http://data.press.net/ontology/stuff/Person	2
http://schema.org/Person	3
http://www.w3.org/ns/person#Person	4
http://www.ontotext.com/proton/protontop#Person	5
http://omv.ontoware.org/2005/05/ontology#Person	6
http://bibframe.org/vocab/Person	7
http://iflastandards.info/ns/fr/frbr/frbrer/C1005	8
http://models.okkam.org/ENS-core-vocabulary.owl#person	9
hhttp://swat.cse.lehigh.edu/onto/univ-bench.owl#Person	9

knowledge" and two considered themselves to have "Strong knowledge" in ontology engineering on a 5-point Likert-Scale from "Expert knowledge" to "No Knowledge". All of the evaluators have developed ontologies before and some are authors of widely cited ontologies. To reduce the number of classes our ten judges had to score for a given query term (for some query terms a naïve string search returns more than 400 results) we asked two experts to pre-select relevant URIs. The experts were asked to go through all resources that matched a query through a naïve string search and evaluate if the URI is either "Relevant" or "Irrelevant" for the given query term. We asked the two experts to judge URIs as "Relevant" even when they are only vaguely related to the query term, i.e. increasing the false positive ratio.

We developed an evaluation tool which allowed our experts to pose a keyword query for the given term that retrieves all matching ontology classes in the search space. Since keyword queries where the intended meaning of the query is unknown are still the prevalent form of input in Semantic Search [16] and since the meaning of the search terms derived from our real query log was also unknown, we needed to establish the main intention for each of our query terms. We used the main definition from the Oxford dictionary for each term and included it in the questionnaire for our judges. We then asked our ten human experts to rate the relevance of the results to each of the 10 query terms from Table 2 according to their relevance to the definition of the term from the Oxford dictionary. After submitting the keyword query, each evaluator was presented with a randomly ordered list of the matching ontology classes in the search space to eliminate any bias. For each result we showed the evaluator, the `URI`, the `rdfs:label` and `rdfs:comment`, the properties of the class and its super-classes and sub-classes. A judge could then rate the relevance of the class with radio buttons below each search result on a 5-point Likert scale with values "Extremely Useful", "Useful", "Relevant", "Slightly Relevant" and "Irrelevant". There was no time restriction for the judges to finish the experiment. We assigned values from 0-4 for "Irrelevant"-"Extremely Useful" for each score and performed a hypothesis test on the average scores per evaluator with a H_0 $\mu = 2$ against H_1 $\mu <> 0$. This resulted in a p-value of 0.0004, a standard error of mean of 0.144 and a 95% confidence interval for the mean score of (0.83,1.49), indicating there is a strong evidence that the average scores per evaluator are not 2 which would indicate a randomness of the scores. We also asked our ten evaluators to score 62 random response URIs for the ten queries again two months after we performed our initial experiment. The average scores of the ten evaluators for these URIs had a correlation coefficient of 0.93, indicating that in average, the scores of the participants in the second study were highly correlated to the scores in the first study.

Table 3 shows the ideal ranking for the query "Person" as derived from the median relevance scores from our ten experts. For ties we considered the resource with the more consistent relevance scores (i.e. the lower standard deviation) as better ranked. Not all ties could be resolved in this way as can be seen for rank No. 9.

3.4 Evaluation and Performance Measures

We consider three popular metrics from the information retrieval community, precision at k (P@k), mean average precision (MAP), and normalized discounted cumulative gain (NDCG). Since we asked our judges to assign a non-binary value of relevance (on a 5-point Likert scale), we converted these values to a binary value for all those metrics that consider a binary notion of relevance. We chose a resource as being relevant to the query term if the relevance score is equal or higher than the average value on the 5-point Likert scale. Changing this cut off value to the right or to the left of the average changes the overall precision of the result. However, the relative performance of the algorithms remains the same.

Precision@k: We are calculating precision at k (P@k) for a k value of 10. P@k in our experiment is calculated as:

$$p@k = \frac{number\ of\ relevant\ documents\ in\ top\ k\ results}{k}$$

Average Precision: The average precision for the query Q of a ranking model is defined as:

$$AP(Q) = \frac{\sum_{i=1}^{k} rel(r_i) * P@i}{k}$$

where *rel(r_i)* is 1 if r_i is a relevant resource for the query Q and 0 otherwise, *P@i* is the precision at i and k is the cut off value (i.e. 10 in our experiment). MAP is defined as the mean of AP over all queries run in this experiment and is calculated as:

$$MAP = \frac{\sum_{Q \in \mathbb{Q}} AP(Q)}{|\mathbb{Q}|}$$

Normalize Discounted Cumulative Gain (NDCG): NDCG is a standard evaluation measure for ranking tasks with non-binary relevance judgement. NDCG is defined based on a gain vector G, that is, a vector containing the relevance judgements at each rank. Then, the discounted cumulative gain measures the overall gain obtained by reaching rank k, putting more weight at the top of the ranking:

$$DCG(Q) = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(1+i)}$$

To compute the final NDCG, we divide DCG by its optimal value iDCG which puts the most relevant results first. iDCG is calculated by computing the optimal gain vector for an ideal ordering obtained from the median of the user assigned relevance scores.

Table 4. MAP

	Person	Name	Event	Title	Loc.	Addr.	Music	Org.	Author	Time
boolean	0.17	0.00	0.23	0.02	0.00	0.66	0.39	0.02	0.08	0.44
tf-idf	**0.75**	0.44	**0.82**	0.51	**0.73**	0.89	**0.48**	0.70	0.28	0.53
BM25	0.19	**0.74**	0.03	0.40	0.08	0.49	0.18	0.32	**0.62**	0.00
vector-space	0.06	0.00	0.19	0.08	0.00	0.58	0.18	0.00	0.01	0.00
pagerank	0.19	0.38	0.55	**0.70**	0.63	0.18	0.04	0.29	0.49	**0.77**
class-match-measure	0.00	0.00	0.00	0.40	0.00	0.35	0.18	0.00	0.02	0.00
density-measure	0.30	0.00	0.08	0.11	0.00	0.50	0.11	0.00	0.07	0.00
semantic-similarity	0.00	0.00	0.00	0.40	0.00	0.35	0.18	0.00	0.00	0.00
between-measure	0.69	0.23	0.40	0.36	0.55	**0.99**	0.14	**0.80**	0.14	0.66

Table 5. NDCG

	Person	Name	Event	Title	Loc.	Addr.	Music	Org.	Author	Time
boolean	0.06	0.00	0.16	0.11	0.00	0.44	0.22	0.07	0.07	0.15
tf-idf	**0.29**	0.20	**0.46**	**0.27**	**0.32**	0.57	**0.39**	0.32	0.15	**0.30**
BM25	0.07	**0.42**	0.02	0.13	0.07	0.32	0.16	0.19	0.14	0.00
vector-space	0.12	0.00	0.06	0.10	0.00	0.36	0.16	0.00	0.01	0.00
pagerank	0.14	0.18	0.28	0.21	0.15	0.18	0.17	0.22	0.11	0.14
class-match-measure	0.00	0.00	0.00	0.15	0.00	0.17	0.16	0.00	0.05	0.00
density-measure	0.25	0.00	0.07	0.13	0.00	0.27	0.19	0.00	0.04	0.00
semantic-similarity	0.00	0.00	0.00	0.15	0.00	0.17	0.16	0.00	0.03	0.00
between-measure	0.22	0.18	0.17	0.24	0.31	**0.69**	0.15	**0.59**	**0.19**	0.19

4 Results

Table 4 and 5 show the MAP and the NDCG scores for all ranking models for each query term, whereas Fig. 1 shows the P@10, MAP, DCG, NDCG scores for each of the eight ranking models on all ten queries. For P@10 and MAP, tf-idf is the best performing algorithm with betweenness measure as the second best and PageRank as the third best. In terms of the correct order of top k results, we found again tf-idf as the best performing algorithm, with betweenness measure and PageRank as the second and third best, respectively.

4.1 Results Analysis

From the results of this experiment it can be seen, somehow surprisingly, that content-based models (i.e. tf-idf and BM25) outperform the graph-based ranking models for most queries. Overall, seven out of ten times, the content-based models achieve a better or equal to the highest NDCG for all ranking algorithms.

However, although tf-idf achieved the highest mean average precision value of 0.6 in our experiment, it is still far from an ideal ranking performance. That is, because the philosophy of tf-idf works well for the tf part, but not so for the idf part when ranking resources in ontologies. The intuition behind tf-idf is that if a word appears frequently in a document, it is important for the document and is given a high score (i.e. tf value), but if it appears in many documents, it is not a unique identifier and is given a low score (i.e. idf value). In ontologies, a resource that is reused across many ontologies is a popular and relatively more important resource in the ontology and the corpus. Therefore, in our experiment, tf-idf successfully ranks a resource high in the result set if that resource is the central concept of the ontology (i.e. it is assigned

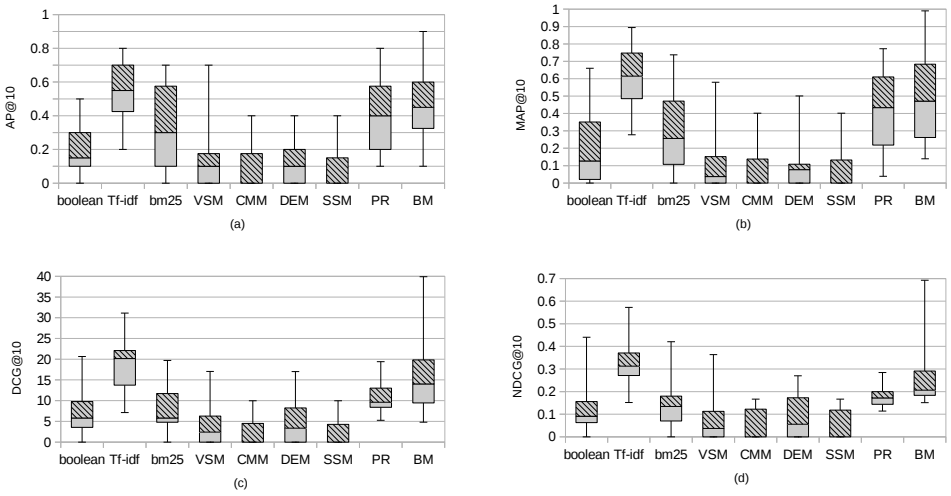

Fig. 1. Effectiveness of Ranking Model

a high tf value). However, if a resource is also popular among the corpus, it is scored down for the idf value. For example, `http://xmlns.com/foaf/0.1/Person` has the highest tf value (i.e. 0.589) of all concepts in the FOAF ontology, but since it is also the most popular concept in our corpus appearing in total in 162 distinct ontologies, it does not appear among the top ten results of tf-idf.

Since BM25 is a cumulative relevance score for an ontology rooted in the tf and idf values of a matched resource, the limitations of tf-idf are implicit in BM25 as well. However, BM25 ranks concept specific ontologies higher in the result set for a query term that matches to that particular concept. The reason is that for a specific ontology, the query term matches to one of the important resource and many of its attached resources. All these matched resources sum up to a higher BM25 score for that ontology. For example, for the "Name" query, BM25 ranks all resources in the GND ontology[10] higher, since this ontology defines different types of Names. All these types of names are important concepts of this ontology and finally leverage the BM25 score for the GND ontology.

The vector space model did not perform well for any query. The main reason is that the vector space model considers tf-idf values of resources as well as query term/s. The idf value for a query term is calculated by considering the idf values of all the resources in the corpus that matched the query term. Therefore, the effect of the wrong assumptions for the idf values doubles for the vector space model.

PageRank ranks resources according to their popularity, that is why it performs, for example, well in ranking highly the "Person" concept in the FOAF ontology as it is a widely used ontology that is imported by many other ontologies. However, considering popularity in the corpus as the only factor for ranking ontologies sometimes results in poor precision and recall. e.g. `http://www.loria.fr/~coulet/ontology/sopharm/version2.0/disease_ontology.owl#DOID_4977` with the label "other road accidents injuring unspecified person" is one of the popular resources in our corpus but not at all relevant for the "Person" concept. Still, PageRank assigns it a higher rank based on its popularity in the corpus. The performance of the PageRank algorithm could be significantly improved if it also takes the data for a

[10] `http://d-nb.info/standards/elementset/gnd#`

given ontology into consideration (as is done in Semantic Search engines). Instead of using only the import statement as the measure of popularity, the links from data will give higher weights to resources in ontologies for which there exists data across multiple domains.

As expected, the class match measures is the least precise algorithm in the experiment. Since the algorithm ranks an ontology only on the basis of the label of the matched resources within that ontology, an ontology with single or zero exact matched labels and many partial match labels gets a higher relevance score than those ontologies where few concepts are relatively more important. Secondly, assigning the same weight to partially matched labels is problematic. For example, for the query "Address" two partially matched resources "Postal address"[11] and "Email address of specimen provider principal investigator"[12] are obviously not equally relevant to the address definition provided in our user study. However, CMM uses equal weights for both of these resources while computing the relevance score of their corresponding ontologies.

The density measure model performs relatively poorly, because it assigns high weights for super-class and sub-class relations. The intention is that the further specified a resource is in an ontology the more important it is. However, in our study the density measure model always favours upper level ontologies or highly layered ontologies, where many subclasses and super classes are defined for a resource (e.g. OBO ontologies), irrespective of its relevance to the query term.

The semantic similarity measure model considers the proximity of matched resources in an ontology. Although this metrics can be useful when considering similarity among the matched resources of two or more query terms of a multi-keyword query, it performs poorly on single word queries. As mentioned earlier, users seem to be not using multi-keyword queries in ontology search yet and thus the semantic similarity measure appears to be not particularly useful for ranking resources in ontologies.

The betweenness measure performs better than all other graph-based ranking models because it calculates the relative importance of the resource to the particular ontology. A resource with a high betweenness value is the central resource of that ontology [1], which means that the resource is well defined and important to the ontology. Further, the betweenness measure performs well even with resources that are irrelevant to the query term if they are not central resources of that ontology, as their score will not contribute greatly to the cumulative relevance score for the ontology. For example, irrelevant resources such as "dislocation" for the query "location" do not appear high in the ranking of the betweenness measure, because all resources with the label including "dislocation" are not central concepts in the ontology where they are defined.

A general observation that can be made is that all ranking models other than tf-idf ignore the relevance and importance of a resources to the query when assigning a weight to a particular ontology for a given query term. This is more prominent for graph-based approaches, where the cumulative ranking score for an ontology is computed based on all the relevant terms of that ontology for this query. An ontology that has more matched URIs to the query term gets a higher weight than an ontology that has few or only a single relevant resource in the ontology. For example, `http://www.ontologydesignpatterns.org/cp/owl/participation.owl#`

[11] `http://purl.obolibrary.org/obo/IAO_0000422`
[12] `http://purl.obolibrary.org/obo/OBI_0001903`

`Event` with the label "event" is ranked "Extremely useful" to "Useful" for the query "event" by our human experts. However, since this is the only relevant resource in the ontology and it is a small ontology, none of the graph-based models ranked this URI among the top ten resources.

5 Recommendations

Based on the analysis of our experiment we put forward the following four recommendations that we believe could significantly improve the performance of the different ranking algorithms.

1. **Intended type vs. context resource:** We believe that differentiating the intended type from the context resource of a URI has a positive impact on the performance of all ranking models. For example, for a resource in the GND ontology[13] with the label "Name of the Person", "Name" is the intended type, whereas "Person" is the context resource. This resource URI appears in the search results for both, the "Person" and the "Name" query term in our experiment. The human experts ranked this resource on average from "Extremely useful" to "Useful" for the "Name" query term and only "Slightly useful" for the "Person" query. However, all the ranking algorithms assigned an equal weight to this resource while calculating ranks for either of the two query terms. The performance of the ranking models could be improved if they either only consider those resource URIs whose intended type is matching the queries intended type or if they assign a higher weight to such URIs as compared to the ones where the query terms' intended type matches only the context resource of that URI.
2. **Exact vs. partial matches:** As identified by Alani et al. [1] exact matching should be favoured over partial matching in ranking ontologies. Whereas the class match measure model assigns a value of 0.6 to exact matches and 0.4 to partial matches, all other algorithms consider partial and exact matched resources equally. For example, for the query "Location", results that include "dislocation" as partial matches should not be considered, since the word sense for location and dislocation are different. Instead of assigning static weight factors, we believe that other means of disambiguation between the actual meaning of the query term and of the resource URI can significantly improve the performance of the algorithms. Wordnet [12] or a disambiguation at the time of entry of the query term could be efficient methods for this purpose.
3. **Relevant relations vs. context relations:** For the graph-based ranking models that calculate the relevance score according to the number of relationships for the resource within that ontology (i.e. density measure and betweenness measure), direct properties, sub-classes and super-classes of a class have to be distinguished from relations (i.e. properties) that are very generic and are inferred from its super-classes. For example, the class "email address"[14] from one of the OBO ontologies has properties like "part of continuant at some time", "geographic focus', "is about", "has subject area", "concretized by at some time", "date/time value" and "keywords". However, not all of these properties are actually relevant to the concept "email address".
4. **Resource relevance vs. ontology relevance:** All ranking models discussed in this study (except tf-idf), rank ontologies for the query term by considering

[13] `http://d-nb.info/standards/elementset/gnd#NameOfThePerson`
[14] `http://purl.obolibrary.org/obo/IAO_0000429`

all matched resources from a given ontology against the query term. This results in a global rank for the ontology and all the resources that belong to that ontology share the same ontology relevance score. Therefore, in a result set, many resources hold the same relevance score. While ordering resources with the same relevance score from the ontology, the ranking models lack a mechanism to rank resources within the same ontology. We believe that the tf value of the resource could be a good measure to assign scores to resources within an ontology. Therefore, while ranking all the resources of an ontology, the tf value can be used to further rank resources that belong to the same ontology. Another solution could be to compute individual measures (all measures other than tf-idf) for each resource, independent of how many other matched resources there are in the same ontology.

6 Conclusion

This paper represents, to the best of our knowledge, the first systematic attempt at establishing a benchmark for ontology ranking. We established a ground truth through a user study with ten ontology engineers that we then used to compare eight state-of-the-art ranking models to. When comparing the ranking models to the ideal ranking obtained through the user study we observed that content-based ranking models (i.e. tf-idf and BM25) slightly outperform graph-based models such as betweenness measure. Even though content-based models performed best in this study, the performance is still inferior to the performance of the same models on ranking documents because of the structural differences between documents and ontologies. We put forward four recommendations that we believe can considerably improve the performance of the discussed models for ranking resources in ontologies. In particular:

- ***Determine the intended type of a resource:*** A resource should only match a query if the intended type of the query matches the intended type of the resource.
- ***Treat partial matches differently:*** Instead of treating partial matches of the query and a resource similar to exact matches or assigning a static weight factor, the models should consider other means of disambiguating the actual meaning of the query when matching it with a resource.
- ***Assign higher weight to direct properties:*** Instead of considering all relations for a class equally when calculating the centrality score in graph-based models, the models should consider assigning a higher score to relations that describe the class directly.
- ***Compute a resource relevance:*** Additionally to computing a relevance score for an ontology as a whole, all ranking models should be changed so that they also compute a score for individual resources within the ontology.

In conclusion, we believe that with few modifications several of the tested ranking models can be significantly improved for the task of ranking resources in ontologies. We also believe that the proposed benchmark suite is well-suited for evaluating new ranking models. We plan to maintain, improve and extend this benchmark, in particular by adding further queries and updating the ontology collection as new ontologies become available. We expect that this will motivate others to produce tailor-made and better methods for searching resources within ontologies.

References

1. H. Alani, C. Brewster, and N. Shadbolt. Ranking Ontologies with AKTiveRank. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 5–9. 2006.

2. M. d'Aquin and H. Lewen. Cupboard — A Place to Expose Your Ontologies to Applications and the Community. In *Proceedings of the 6th European Semantic Web Conference*, pages 913–918, Berlin, Heidelberg, 2009. Springer-Verlag.
3. M. d'Aquin and E. Motta. Watson, More Than a Semantic Web Search Engine. *Semantic Web*, 2(1):55–63, 2011.
4. L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: A Search and Metadata Engine for the Semantic Web. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 652–659, New York, NY, USA, 2004. ACM.
5. M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic Search Meets the Web. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 253–260, Washington, DC, USA, 2008.
6. A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. A theoretical framework for ontology evaluation and validation. In *Proceedings of the 2nd Italian Semantic Web Workshop*, volume 166 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
7. N. Guarino and C. Welty. Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45:61–65, 2002.
8. A. Harth, J. Umbrich, A. Hogan, and S. Decker. YARS2: A Federated Repository for Querying Graph Structured Data from the Web. In *Proceedings of the 6th International Semantic Web Conference*, pages 211–224, Berlin, Heidelberg, 2007.
9. A. Hogan, A. Harth, and S. Decker. Reconrank: A scalable ranking method for semantic web data with context. In *Proceedings of the 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.
10. A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9"(4):365–401, 2011.
11. A. Lozano-Tello and A. Gomez-Perez. ONTOMETRIC: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18, 2004.
12. G. A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, Nov. 1995.
13. N. F. Noy and M. d'Aquin. Where to Publish and Find Ontologies? A Survey of Ontology Libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11(0), 2012.
14. N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, and M. A. Musen. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 2009.
15. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.
16. J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 771–780, New York, NY, USA, 2010. ACM.
17. S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995.
18. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
19. G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
20. E. Thomas, J. Z. Pan, and D. Sleeman. Ontosearch2: Searching ontologies semantically. In *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
21. G. Tummarello, R. Delbru, and E. Oren. Sindice.Com: Weaving the Open Linked Data. In *Proceedings of the 6th International Semantic Web Conference*, pages 552–565, 2007.
22. P.-Y. Vandenbussche and B. Vatant. Linked Open Vocabularies. *ERCIM news*, 96:21–22, 2014.