

# Ontology Search: Finding the Right Ontologies on the Web

Anila Sahar Butt  
«Supervised by Armin Haller»  
The Australian National University  
Canberra, ACT, Australia.  
anila.butt@anu.edu.au

## ABSTRACT

With the recent growth of Linked Data on the Web, there is an increased need for knowledge engineers to find ontologies to describe their data. Only limited work exists that addresses the problem of searching and ranking ontologies based on keyword queries. In this proposal we introduce the main challenges to find appropriate ontologies, and preliminary solutions to address these challenges. Our evaluation shows that the proposed solution performs significantly better than existing solutions on a benchmark ontology collection for the majority of the sample queries defined in the benchmark.

## Keywords

Ontology Search; Ontology Ranking; Ontology Benchmark

## 1. PROBLEM

Ontology is considered as a means of a shared conceptualization of a domain knowledge. The growth in Linked Data coupled with the widespread use of ontologies in vertical domains (e.g. bioinformatics, e-commerce, internet-of-things etc.) highlights an increasing need of ontologies. However, the development of an ontology from scratch is a resource intensive process. The process of reusing existing ontologies is cost-effective because it saves engineering efforts to build an ontology. It also produces high-quality ontologies because using the same ontology or ontological terms are assumed to hold the same view upon the modelled universe of discourse. Another major benefit of reusing existing ontologies is its potential to enable and facilitate data interoperability on both the syntactic and the semantic level. However, the potential to “reuse” ontologies is hampered by the fact that it is hard to find the right ontology for a given use case.

As a result ontology search is intensely pragmatic and strenuous. The ontological data is restricted in size due to the limited number of published ontologies. Consequently, the ranking of matched results becomes the core concern of

ontology search rather than its efficiency. Moreover, in ontology search keyword queries are still the preferred method to find concepts and relations in the registered ontologies [2]. The search results are a match of a search term with more expressive concepts, relations or ontology descriptions. There may exist many ontologies that contain concepts and relations with their labels match the keyword query, however they have been described in ontologies differently mainly in terms of their: (i) **perspective** - A concept may be defined in different perspectives e.g., The person class is defined in many ontologies, but the “foaf”<sup>1</sup> ontology captures the social aspects of person, whereas the “appearance” ontology models the natural attributes of a person i.e. weight, height, and nature, (ii) **levels of detail** - the concepts are defined in the same perspective in different ontologies, but in different levels of detail i.e. abstract or detailed, and (iii) **extension** - the concepts are defined in one ontology and then extended in another ontology. The problem is how to find and order many matched results for a keyword search to satisfy a user information need.

There has been some previous work, for example [1, 11, 10], to tackle the problem of finding and ranking ontologies. More recently, a dedicated ontology search engine has emerged [14], however there are three major issues with existing approaches: (1) The ranking algorithms they use are based only on document-ranking algorithms. Moreover, most of the ranking techniques in existing ontology libraries and search engines only consider the popularity of terms in the ontology corpus, often using the PageRank algorithm, which although effective in some cases [2] hinders the visibility of newly emerged well defined ontologies. (2) Most of the ontology search systems retrieve ontological terms (concepts and relations) and some provide ontology search based on some keywords. However, almost all ontology libraries and search engines do not facilitate the task of ontology retrieval for a use-case where a user looks for an ontology that models all or some of the concepts and relations. BioPortal [11] provides an opportunity to find an ontology based on the text description, however it is a domain dependant ontology library and does not deal with all types of ontologies. A general solution is required for ontology search based on text description or one or more keywords (3) All ontology libraries and search engines claim to perform best. However, there are no standard means to measure the performance to verify that they perform up to the developer’s performance claim. This is evidently important to facilitate the user who needs to select a library or search engine according to her

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author’s site if the Material is used in electronic media.  
WWW 2015 Companion, May 18–22, 2015, Florence, Italy.  
ACM 978-1-4503-3473-0/15/05.  
<http://dx.doi.org/10.1145/2740908.2741753>.

<sup>1</sup><http://xmlns.com/foaf/spec/>

application requirement, or for the performance evaluation of new ontology search approaches. On the basis of above discussion the following are the principle research questions for this research:

**RQ1: Ontology Ranking Model.** How to rank relevant resources and ontologies for keyword queries?

**RQ2: Ontology Retrieval Framework.** How to efficiently find the most relevant resources, and ontologies that cover one or more resources users are interested in?

**RQ3: Evaluation Framework for ontology retrieval approaches.** How to evaluate the newly emerging ontology libraries and search engines in comparison to existing ones?

## 2. STATE OF THE ART

The Linked Open Vocabularies (LOV) search engine, initiated in March 2011, is to the best of our knowledge, the only purpose-built ontology search engine available on the Web. It uses a full-text inverted index and a ranking algorithm based on the term popularity in Linked Open Data (LOD) and in the LOV ecosystem [14]. There are also some ontology libraries available that facilitate the locating and retrieving of potentially relevant ontology resources [10]. Some of these libraries are domain-specific such as the Open Biological and Biomedical Ontologies library<sup>2</sup> or the BioPortal [11], whereas others are more general such as OntoSearch [12] or the TONES Ontology Repository<sup>3</sup>. However, as discussed by Noy & d'Aquin [10] only few libraries support a keyword search, only one (Cupboard) supports a ranking of ontologies based on a keyword query using an information retrieval algorithm (i.e. tf-idf), and none support the ranking of resources within these ontologies.

Semantic Search engines such as Swoogle [6] (which was initially developed to find ontologies only), Sindice.com [13], Watson [5], or Yars2 [8] do allow a search of ontology resources through a keyword query. The ranking in these search engines follows traditional link-based ranking methods [9], in particular adapted versions of the PageRank algorithm, 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]. AKTiveRank [1], finds and ranks ontologies based on how well they cover specified search terms. Falcon [4] is a popularity-based scheme to rank concepts and ontologies. Other strategies, mainly based on methods proposed in the information retrieval community, are employed in Semantic Search [7], but what all these methods have in common is that they are targeted to rank instances, but do not work well for ranking concepts and properties in ontologies [6, 1]. Another related approach is presented in [15] that identifies the most important concepts and relationships from a given ontology. However, the approach does not support ranking concept that belong to multiple ontologies.

## 3. CBRBENCH: CANBERRA ONTOLOGY RANKING BENCHMARK

This work is conducted to design a benchmark to evaluate the effectiveness of ranking models. We have implemented eight ranking algorithms, four of which have been proposed

<sup>2</sup><http://www.obofoundation.org/>

<sup>3</sup><http://owl.cs.manchester.ac.uk/repository/>

by the information retrieval community: Tf-Idf, BM25, Vector Space Model (VSM) and PageRank (PR), 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 URLs 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, and 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. The contribution of this work are:

- 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 details of this work are presented in [2]. We are still looking into ways to extend the benchmark to evaluate search techniques efficiency and effectiveness in recommending an ontology covering user specified terms.

## 4. DWRANK: A DUAL WALK BASED RANKING MODEL

We proposed a ranking model that assigns a rank score to each concept defined in ontology corpus. Our proposed ranking model characterises two features of a concept to determine its rank in a corpus: (1) A concept is more important, if it is a *central concept* to the ontology within which it is defined. (2) A concept is more important, if it is defined in an *authoritative* ontology. On the basis of above assumption we define two scores for each concept in ontology corpus. *Hub score*, a measure of the *centrality of a concept*, i.e. the extent that the concept is **related** to the domain for which the ontology is formalised. *Authority score* is a measure of the **authoritativeness** of the ontology. A link analysis algorithm, similar to PageRank, is performed that leverages the ontological structure and semantics to compute these scores. However, the difference between our model and a traditional PageRank-like algorithms is two-fold. Firstly, we perform the link analysis independently on each ontology to find a *hub score* and then only on the whole ontology corpus considering an ontology as a node and inter-ontology relationships (i.e. ontology imports links) as links. Secondly, we differentiate the type of relationship and the direction of the walk varies on the basis of the type of the relationship.

## 4.1 HubScore

The *hub score* is a measure of the centrality of a concept within an ontology. We define a *hub function*  $h(v, O)$  that calculates the *hub score*. The *hub function* is characterised by two features: (1) **Connectivity**: A concept is more central to an ontology, if there are more *relationships* (i.e. links) starting from the concept. (2) **Neighbourhood**: A concept is more central to an ontology, if there is an *relationship* starting from the concept to another central concept.

According to these features, a concept accepts the centrality of another concept based on its forward link concepts (like a **hub**). The *hub function* is therefore a complete reverse of the PageRank algorithm where a node accepts scores from its referent nodes i.e. back link concepts. We adopt a Reverse-PageRank as the *hub function* to find the centrality of a concept within the ontology. An important modification made in linked analysis is made by introducing *weak nodes*, these are artificial concepts  $V(O)$  in the ontology that act as a sink for every *data-type relationship*. We label these concepts with the data type relationship label. After incorporating *weak nodes* notions, Eq. 1 reflects the complete feature of our *hub function*.

$$h_k(v, O) = \frac{1 - \alpha}{|V|} + \alpha * \sum_{v_i \in C_{SFLinks}(v, O) \cup C_{WFLinks}(v, O)} \frac{h_{k-1}(v_i, O)}{|C_{BLinks}(v_i, O)|} \quad (1)$$

In Eq. 1,  $h_k(v, O)$  is hub score of concept  $v$  of ontology  $O$  at  $k^{th}$  iteration,  $C_{SFLinks}(v, O)$  is a set of *strong forward link concepts*,  $C_{WFLinks}(v, O)$  is a set of *weak forward link concepts* and  $C_{BLinks}(v_i, O)$  are concepts linked to  $v_i$  through backward links. We normalise the hub scores of each concept  $v$  within an ontology  $O$  through the **z-score** statistical measure after the last iteration of the *hub function*.

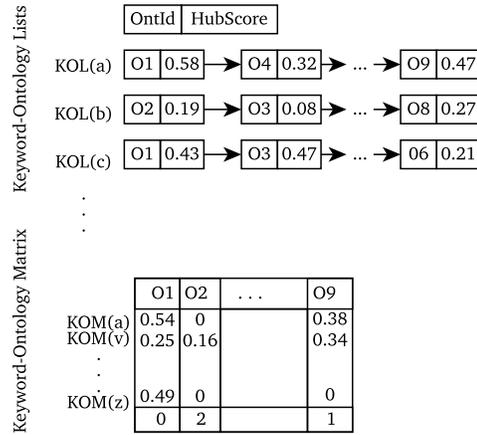
## 4.2 AuthorityScore

The *authority score* is the measure of the authoritativeness of a concept within an ontology. The *authoritativeness of a concept* depends upon the *authoritativeness of the ontology* within which it is defined. Therefore, we define the *authority function*  $a(O)$  to measure the *authority score* of an ontology. Our *authority function* is characterised by the following two features: (1) **Reuse**: An ontology is more authoritative, if there are more *inter-ontology relationships* (i.e import links) ending at the ontology. (2) **Neighbourhood**: An ontology is more authoritative, if there is an *inter-ontology relationship* starting from an authoritative ontology to the ontology.

The PageRank is adopted as the *authority function*, whereby each ontology is considered a node and *inter-ontology relationships* are considered links among nodes. Eq. 2 formalise the *authority function* which computes the authoritativeness of  $O$  at the  $k^{th}$  iteration.

$$a_k(O) = \frac{1 - \alpha}{|O|} + \alpha \sum_{O_i \in O_{BLinks}(O)} \frac{a_{k-1}(O_i)}{|O_{FLinks}(O_i)|} \quad (2)$$

In Eq. 2,  $O_{BLinks}(O)$  is a set of *back link ontologies* and  $O_{FLinks}(O)$  is a set of *forward link ontologies*. The definition of  $O_{FLinks}(O)$  (resp.  $O_{BLinks}(O)$ ) is similar to  $C_{FLinks}(v, O)$  (resp.  $C_{BLinks}(v, O)$ ), however, the links are inter-ontology relationships. Similar to the *hub score*, we



**Figure 1: Keyword-Node Lists and Keyword-Node Matrix for Ontology Retrieval**

also compute the **z-score** of each ontology after the last iteration of *authority function*.

## 4.3 DWRank Score

We define two versions of DWRank function.

1- **DWRank Linear Model** : We define the *DWRank*  $R_{(v, O)}$ , as a function of the *text relevancy*, the *normalised hub score* and the *normalised authority score*. The function is described as a quantitative metric for the overall relevance between the query  $Q$  and the concept  $v$ ; and the concept *hub and authority score* as follows:

$$R_{(v, O)} = F_V(v, Q) * [w_1 h(v, O) + w_2 a(O)]$$

$$F_V(v, Q) = \sum_{q \in Q} f_{ss}(q, \phi(q_v)) \quad (3)$$

In Eq. 3,  $w_1$  and  $w_2$  are the weights for the *hub function* and the *authority function*.  $F_V(v, Q)$  aggregates the contribution of all matched words of a node  $v$ , in an ontology  $O$ , to the query keywords  $q \in Q$ .  $f_{ss}$  returns a binary value : it returns 1 if  $q$  has a match  $\phi(q_v)$  in  $v$ , and 0 otherwise. The metric favours the nodes  $v$  that are semantically matched to more keywords of the query  $Q$ .

2- **DWRank Learning to Rank Approach** : In second version, DWRank model is learnt using a feature set. Along with the core features of DWRank, i.e. *text relevancy*, *hub score* and *authority score*, two extra features *max\_hub score* and *min\_hub score* are introduced while training the ranking model to normalize the *hub score* across ontologies. (i) **max\_hub score**: For a concept  $v$ , max\_hub score for  $v$  is the maximum score of any concept  $v'$  in the ontology where  $v \in V(O)$  and  $v' \in V(O)$ . (ii) **min\_hub score**: For a concept  $v$ , min\_hub score for  $v$  is the minimum score of any concept  $v'$  in the ontology where  $v \in V(O)$  and  $v' \in V(O)$ . Details of this version are presented in Sec. 7.1.

## 5. ONTOLOGY SEARCH FRAMEWORK

### 5.1 Concept Retrieval Framework

The concept retrieval framework is composed of two phases. The first phase is an offline phase where two indices, i.e.

*ConHubIdx* and *OntAuthIdx*, are constructed for the whole ontology corpus. The second phase is an online query processing phase where a query is evaluated and the *top-k* concepts are returned to the user. Details about complete framework are presented in [3]. An overview is as following:

**Offline Ranking and Index construction:** The framework first constructs a *ConHubIdx* on all concepts and *OntAuthIdx* on all ontologies in the ontology corpus  $\mathcal{O}$ . The *ConHubIdx* maps each concept of an ontology to its corresponding *hub score* and the *OntAuthIdx* maps each ontology to its precomputed *authority score* and ontology Id.

**Online Query Processing:** Upon receiving a query  $\mathcal{Q}$ , the framework extracts the *candidate result set*  $C_Q = \{(v_1, O_1), \dots, (v_i, O_j)\}$  including all a matches that are semantically similar to  $\mathcal{Q}$  by querying the ontology repository. The *hub score* and *authority score* for all  $(v, O) \in C_Q$  are extracted from the corresponding indices as  $H(C_Q)$  and  $A(C_Q)$  lists. A ranked list  $R(C_Q)$  of a candidate result set is computed from  $H(C_Q)$  and  $A(C_Q)$  along with the text relevancy measure.

## 5.2 Ontology Retrieval Framework

We propose a storage and indexing mechanism for ontologies to retrieve them efficiently and effectively. For this purpose, each ontology in the ontology collection is parsed to build two additional indexes.

1. **Keyword-ontology-list:** A linked list is created for each keyword after stemming (excluding stop-words) that appears in the label of resources of ontologies. For a keyword  $w$ ,  $KOL(w)$  denotes the list of ontologies that cover a concept or relation related to this keyword. Each node in the linked-list contains two fields i.e. ontology id and hub-score of a matched resource of the ontology as shown in Fig. 1. If more than one resource of the ontology matches to the keyword then only maximum hubscore for any matched resource is considered for that ontology.

2. **Sorted-string-index:** An tree-based sorted string index is created on the ontology collection. Each entry in the tree contains two fields: (i) a keyword and (ii) starting address of the corresponding *keyword-ontology-list*.

A query is evaluated on the basis of the aforementioned indexes. The query keywords, corresponding to some concepts that a user is looking for are searched into **sorted string index** after stemming. Each matched keyword in the node of tree returns the starting address of **keyword-ontology-list**. An in-memory **Keyword-ontology-matrix** is created to get the matched ontologies in a sequence. For each query keyword **Keyword-ontology-list** is retrieved and a new row is introduced in the matrix. From the **keyword-ontology-list**, nodes are accessed sequentially and for each node a column is introduced, if a column for the ontology-id does not exist, in the matrix. DWRank score for the ontology for the query keyword is computed using the *hub\_score* saved in the second field of the node and *authority\_score* from *AuthOntIdx* and added as the value of the corresponding cell (keyword-ontology). If keyword does not exist in an ontology corresponding to the column in **Keyword-ontology-matrix** then 0 is added for that cell. All ontologies corresponding to the **Keyword-ontology-matrix** have at-least one or more query keywords as label of its resources. Once the **Keyword-ontology-matrix** is built for a query zeros are counted in each column. The ontologies with minimum count of zero covers maximum query-keywords and get the highest order in the resultset. If more than one ontologies

have the same number of zeros, their DWRank score is added for each keyword and ranked in the increasing order of rank score.

## 6. EVALUATION PLAN

Our approach will be evaluated in terms of its efficiency and effectiveness. The efficiency of the approach provides a measure of its scalability and effectiveness is measure of quality of retrieved results.

- Effectiveness of the ranking model will be evaluated by comparing the results with the state-of-the-art techniques proposed by [2].
- Effectiveness of the ontology retrieval framework will be evaluated by human judges on a 5-point likert scale. Performance metrics such as  $P@k$ ,  $MAP@k$  and  $NDCG@k$  will be computed for the graded results.
- Efficiency of the overall framework will be evaluated in-terms of (i) query processing time (ii) complexity of index construction and updation.
- An empirical study will be conducted to evaluate the usefulness and usability of the entire system in collaboration with OEG<sup>4</sup>.

## 7. CURRENT RESULTS

### 7.1 Effectiveness of DWRank Model

In the first set of experiments, we evaluated the effectiveness of DWRank in comparison with the baseline ranking models.

**Experiment-1: Offline Learning** In this experiment, we study the impact of offline training on the quality of ranking. For the evaluation we implemented two versions of DWRank:

1. **DWRank Fixed Weight Linear Model:** where hub score, authority score and text relevancy are combined in a linear model (i.e. Eq. 3) and the values of weights  $\alpha$ ,  $\beta$  and  $\gamma$  are set to 0.5, 0.5 and 1 respectively.
2. **DWRank with Learning to Rank Approach:** By using LambdaMART, a LTR algorithm, a ranking model is learnt from the hub score, the authority score and the text relevancy along with two deduced features i.e. the *max\_hub* score and the *min\_hub* score.

For **DWRank fixed weight linear model**, we executed the ten sample queries on the ontology collection and retrieved the *top-k* results according to the proposed linear ranking model in Eq. 3. We recorded the  $P@10$ , the  $MAP@10$ , and the  $NDCG@10$ . For evaluation purpose of DWRank with learning to rank approach, **Leave-one-out Cross Validation (LOOCV)** is adopted. Where for  $n$  number of queries we remove the relevance judgement for training examples of one query and train the ranking model on training examples of the remaining  $n - 1$  queries and then we evaluate the performance of the trained model on  $n_{th}$  query. Once the process is repeated for  $n$  queries, the mean performance is computed. We applied LOOCV on the queries and the gold standard; and record the  $P@10$ , the  $MAP@10$ , the  $DCG@10$  and

<sup>4</sup>Ontology Engineering Group - <http://www.oeg-upm.net/>

the NDCG@10. The experimental results show that DWRank where hub and authority scores along with the text relevancy are combined by a model learnt through LTR, performs better than the DWRank fixed weight linear model.

**Experiment-2: Effectiveness of Top-k Search** Next, we compared our results with the baseline for the same dataset with the sample queries. We compare the performance of the **DWRank fixed weight linear model** (so onward referred as DWRank) with the baseline algorithms. The results show that DWRank performs better than the best performing ranking algorithm for most queries. For some of the queries, the P@10 and MAP@10 for DWRank is lower than the other best performing ranking models. However, the maximum average MAP@10 for DWRank on ten queries is 0.80 that is greater than the average of Tf-Idf, the best baseline ranking models, (i.e., 0.55). The MAP@10 of DWRank ranges from 0.65 ~ 1.0 that means the performance of DWRank is more stable on the ontology collection for the sample queries than the baseline ranking models.

## 7.2 Quality of Hub Function

To evaluate the quality of the *hub score* we consider CARRank [15] as a baseline. The reason of comparing the *hub score* quality with the quality of CARRank is two fold: (1) CARRank use a similar approach (i.e. ReversePage Rank), and (2) the performance results in [15] prove it a better approach than other centrality measures e.g. Betweenness Measure[1] and Density Measure[1]. Since the CARRank algorithm and the gold standard are not available online, we implement CARRank in Java and adopted a similar evaluation strategy as presented in [15].

To evaluate the two approaches, we tried to collect ontologies and their top-10 concepts. Four representative ontologies where members of CSIRO<sup>5</sup> are the part of ontology design team were selected. We asked the ontology creators of the four ontologies to list the top 10 central concepts of the ontology they designed. We then compare the reference ranking produced by the ontology creator with the top-10 ranked list generated by HubScore and CARRank. On average the precision of HubScore on the representative ontologies increases by 20% compared to CARRank, the ranked list also seems more meaningful than CARRank.

## 8. CONCLUSION AND FUTURE WORK

Ontology search is relatively less explored and increasingly important for 'ontology reuse' and thus the data interoperability in semantic Web. In this doctoral proposal, we highlight the limitations of ontology search approaches and propose solutions for them. In this stage, the main focus is on investigating the proposed ontology retrieval framework to make exploratory search available on ontologies. We are also trying to bring some improvements in DWRank model and extension of CBRBench. In addition, we plan to verify the feasibility of the proposed method in practice by allowing others researchers to evaluate our approach.

## 9. 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.

<sup>5</sup><http://csiro.au/>

- [2] A. S. Butt, A. Haller, and L. Xie. Ontology search: An empirical evaluation. In *Proceedings of the International Semantic Web Conference*, pages 130–147, Riva del Gara, Italy, 2014.
- [3] A. S. Butt, A. Haller, and L. Xie. Relationship-based top-k concept retrieval for ontology search. In *Knowledge Engineering and Knowledge Management*, pages 485–502. 2014.
- [4] G. Cheng, W. Ge, and Y. Qu. Falcons: searching and browsing entities on the semantic web. In *Proceedings of the 17th international conference on World Wide Web*, pages 1101–1102. ACM, 2008.
- [5] M. d'Aquin and E. Motta. Watson, More Than a Semantic Web Search Engine. *Semantic Web*, 2(1):55–63, 2011.
- [6] 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, USA, 2004.
- [7] 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.
- [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, ISWC'07/ASWC'07*, pages 211–224, Berlin, Heidelberg, 2007. Springer-Verlag.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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*, 2007.
- [13] G. Tummarello, R. Delbru, and E. Oren. Sindice.Com: Weaving the Open Linked Data. In *Proceedings of the 6th International The Semantic Web Conference*, pages 552–565, Berlin, Heidelberg, 2007.
- [14] P.-Y. Vandenbussche and B. Vatant. Linked Open Vocabularies. *ERCIM news*, 96:21–22, 2014.
- [15] G. Wu, J. Li, L. Feng, and K. Wang. Identifying potentially important concepts and relations in an ontology. In *The Semantic Web-ISWC 2008*, pages 33–49. 2008.