Context-driven Concept Search across Web Ontologies using Keyword Queries

Chetana Gavankar
IITB-Monash Research Academy, Mumbai, India
chetanagavankar@gmail.com

Yuan-Fang Li
Monash University, Melbourne, Australia
yuanfang.li@monash.edu

Ganesh Ramakrishnan
IIT Bombay, Mumbai, India
ganesh@cse.iitb.ac.in

ABSTRACT

Concepts in ontologies can be used in many scenarios, including annotation of online resources, automatic ontology population, and document classification to improve web search results. Collectively, tens of millions of concepts have been defined in a large number of ontologies that cover many overlapping domains. The scale, duplication and ambiguity makes concept search a challenging problem. We present a novel concept search approach that exploits structures present in ontologies and constructs *contexts* to effectively filter the noise in concept search results. The three key components of our approach are (1) a *context* for each concept extracted from relevant properties and axioms, (2) *query interpretation* based on the extracted context and (3) result *ranking* using learning to rank algorithms. We evaluate our approach on a large dataset from BioPortal. Our comprehensive evaluation is performed on 2,062,080 concepts and more than 2,000 queries, using two widely-employed performance metrics: normalized discounted cumulative gain (NDCG) and mean reciprocal rank (MRR). Our approach outperforms BioPortal significantly for multitoken queries that make up a large percentage of total queries.

Keywords

Ontology Concept Search, Indexing, Query Interpretation

1. INTRODUCTION

The ever increasing availability of structured data on the web has led to the challenging problem of searching across this data. The Linked Open Data project [8] connect these datasets and ontologies from different domains that range from biomedical, academic to government and social media. The current breed of Semantic Web search engines can be broadly grouped into three categories: (1) those that search for ontologies [7, 6], (2) those that search for individual resources [11, 7] and (3) those that search for concepts that represent a group of individuals [4, 20]. Searching for ontologies is sometimes too coarse-grained because a large ontology may contain hundreds of thousands, or even millions of concepts. On the other hand, searching for individual resources could be too fine-grained since many resources may be relevant and returning them individually may not be the most useful approach. Searching for concepts forms an ideal middle ground and can be useful in a wide variety of applications such as annotation of online resources, ontology population, and document classification for web search. The majority of these applications require searching for concepts across large overlapping ontologies. In certain domains such as life sciences, there are many overlapping domain ontologies that primarily contain concepts and properties that describe and link concepts. To the best of our knowledge, existing concept search approaches can be primarily divided into two types based on the nature of input queries: SPARQL queries [14] at one end, and keyword queries [7] at the other. The use of SPARQL queries as precise input queries leads to exact results. However, it requires the user to be technically good at writing SPARQL queries and to have knowledge of the schema of the ontologies to be queried. In particular, often the schema might not be known to the user who writes the query. Existing keyword query based approaches [4, 20, 7] typically use standard Information retrieval techniques including tf-idf based search and ranking and popularity based PageRank algorithms. However, these approaches do not capture the context necessary for interpreting queries with multiple keywords.

In this paper, we present a novel context-driven approach of searching for concepts across ontologies using keyword queries. Ontology axioms and class properties are used as contextual features to improve accuracy and assist in disambiguating user queries. The primary technical contributions of our approach are three-fold: (1) Context extraction for each concept based on relevant properties and axioms (2) Query interpretation based on extracted contexts and (3) Ranking search results using *learning to rank* algorithms.

2. RELATED WORK

The Semantic search engines such as Sindice [23], Swoogle [7, 6], Watson [5], BioPortal [18], Falcons [4, 20] enable keyword based search for ontology and the individual resources within them. Current semantic search systems search for ontology, documents and terms on the semantic web whereas we propose to search concepts across ontologies. The search systems use standard information retrieval approaches that are based on tf-idf and PageRank algorithm. Our context driven approach uses lexical co-occurrence measures and direct and indirect relation extraction for query disambiguation. Our approach is optimized for multiple token queries in which, the context plays an important role.

SchemEX [15] is a stream based approach and tool for real

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

K-CAP 2015, October 07-10, 2015, Palisades, NY, USA

© 2015 ACM. ISBN 978-1-4503-3849-3/15/10…$15.00

DOI: http://dx.doi.org/10.1145/2815833.2816958

time indexing and schema extraction of LOD data. Recent work in the area of Semantic Web resources ranking is largely based on adapting and modifying the PageRank algorithm [19]. ReConRank [10] adapts the PageRank and HITS [13, 9] algorithms for Semantic Web data. AKTiveRank [1] ranks ontologies based on how well they cover the specified search terms. The Linked open vocabularies (LOV) [2] search system ranks results based on term popularity in the LOD datasets and in the LOV ecosystem. The paper by Butt *et. al.* [3] present comparison of eight ranking algorithms for searching resources within ontologies. They evaluate the performance of the the traditional IR based ranking algorithms used in semantic web search. The current systems adapt the indexing and pagerank based ranking for semantic search sytems. We make use of the rich ontology structure for indexing and use learning to rank approach for ranking search results.

The work on Top-k exploration of query candidates on (RDF) graph data[22] proposes an intermediate step of converting keyword queries to structured queries. The user needs to select the correct SPARQL queries to finally retrieve search results. Although our indexing techniques appear to be similar to theirs, their approach requires user input to select the appropriate SPARQL query interpretation, while our method internally interprets the keyword query without needing to explicitly generate candidate SPARQL query translations for the keyword query.

3. PROBLEM DEFINITION

Given a multi-word keyword query on a search space of diverse Web ontologies, the goal is to retrieve the relevant concepts with high quality top-k ranking using the context of each concept/class.

Keyword query Information need is represented as query Q. Query Q can consist of m keywords, $Q=\{k_1,k_2,...,k_m\}$.

Search space Let the ontologies be $\mathcal{O}=\{O_1,O_2,...,O_n\}$. The (named) concepts in a given ontology $O_i \in \mathcal{O}$ are represented by $C^i=\{C^i_1,C^i_2,...,C^i_p\}$.

Entities, names & annotations Given an ontology O, an *entity* is a named concept or a named property declared in the ontology. Given an axiom $a \in O$ (logical or annotation), let $\texttt{entities}(a)$ represent the set of entities that appear in a. For an entity e, let function $\texttt{name}(e)$ represent the name of e, and let function $\texttt{annotations}(e)$ represent the values of annotation axioms on e. These annotation axioms include `rdfs:label`, `rdfs:comments`, `rdfs:isDefinedBy`,etc.

Context of a concept The context of a concept is defined as the set of annotation values of the concept and of the entities in relevant axioms in the ontology.

$$Ax_{C_j} = \{a | a \in O \wedge C_j \in \texttt{entities}(a) \wedge \quad (1)$$
$$a \text{ is a } \texttt{SubClassOf} \text{ or } \texttt{EqvClass} \text{ axiom}\}$$

$$Px_{C_j} = \{a | a \in O \wedge C_j \in \texttt{entities}(a) \wedge \quad (2)$$
$$a \text{ is an object/data property axiom}\}$$

$$Context(C_j) = \{\texttt{name}(C_j)\} \cup \{\texttt{annotations}(C_j)\} \cup \quad (3)$$
$$\{\texttt{name}(e) | \forall a \in Ax_{C_j}, e \in \texttt{entities}(a) \wedge$$
$$C_j \in \texttt{entities}(a)\} \cup$$
$$\{\texttt{name}(e) | \forall a \in Px_{C_j}, e \in \texttt{entities}(a) \wedge$$
$$C_j \in \texttt{entities}(a)\}$$

where Ax_{C_j} and Px_{C_j} are sets of class- and property-axioms relevant to C_j, respectively, where Px_{C_j} includes `domain` and `range` axioms. $Context(C_j)$ is the set of names of entities that are relevant to C_j, together with the name and annotation values of C_j.

4. CONCEPT SEARCH FRAMEWORK

Our concept search approach is based on the understanding that the tokens in the user query are interrelated. The query interpretations using these relations are generated in a context-driven approach. The user intention in the search query can be interpreted in two ways. Either generate *implicit* query interpretations such as generative language models [21] or generate *explicit* query interpretations with clearly interpretable search results. In our ontology search setting, we take advantage of the rich structure to generate *explicit* interpretations using the context properties and axioms related to the concepts.

4.1 Context-based Search

In the context based search, we index the context information of a concept via axioms and properties in an ontology. The context information in the index along with co-occurrence information among keywords is used for query interpretations. We evaluate co-occurrence among the keywords in the query using lexical co-occurrence measures explained in detail in Section 4.2. The association among keywords is used for the query interpretation (QI) discussed in Section 4.3. QI evaluates direct and indirect relations among keywords using context of concept and obtains search results. The feature vectors (fv's) are built for the search results and are used to rank the search results using *learning to rank* algorithms. We present an outline of our search framework in Figure 1.

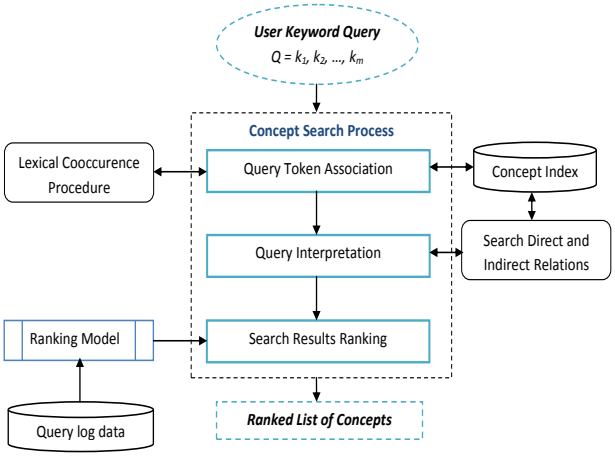

Figure 1: Our search framework

4.2 Query Token Association

We evaluate co-occurrence among keywords using the Pearson's Chi square measure. We can reject null hypotheses that the two words are independent with 95% confidence if the Pearson Chi square > 3.841.

Pearson's Chi square The Pearson's Chi square test compares the observed frequencies $O_{i,j}$ with the frequencies expected $E_{i,j}$ for independence. If the difference between observed and expected is large than we reject the null hypotheses of independence. The χ^2 statistic sums the differences between observed and the expected values in all squares scaled by magnitude of expected frequencies as follows:

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \quad (4)$$

4.3 Query Interpretation (QI)

The query interpretations are generated using contexts of concepts. The co-occurring tokens along with single tokens in the keyword query are used to find direct relations, in which all the keywords in the query are related via properties and axioms relevant to the concept. The indirect relations are found via named class expressions found in equivalent and subclass axiom. We formally define the direct and indirect relation further in this section.

The name and annotation values of class C_j is represented as $L(C_j)$ as defined in Section 3.2. With slight abuse of notation, we use $\texttt{name}(Px_{C_j}) = \bigcup_{a \in Px_{C_j}} \{\texttt{name}(\texttt{entities}(a))\}$ to denote the names of all those entities appearing in property axioms relevant to concept C_j. k' and k'' are two partitioning subsets of keywords in a query, where k' is the set of co-occurring and/or single terms in the query found using co-occurrence measure presented in Section 3.3, and k'' is the set of tokens in the query which are not in k' ($k'' = Q \backslash k'$). The cosine similarity measure is used by the $match(L(C_j),k')$ function to search for concepts L_j where $L(C_j)$ matches k'. Similarly, $match(\texttt{name}(Px_{C_j}),k'')$ uses the same similarity measure to find concepts C_j where $\texttt{name}(Px_{C_j})$ and k'' match.

Direct Relation Direct relation among the query tokens is function of $L(C_j)$ and $\texttt{name}(Px_{C_j})$. Direct relation among tokens k' and k'' is:

$$DR(k',k'') := match(L(C_j),k') \cup match(\texttt{name}(Px_{C_j}),k'') \quad (5)$$

Indirect Relation The indirect relations are formed via relations among tokens in the query via properties of other concept (equivalent-class or subclassOf concept) which does not directly appear in keyword query, but indirectly via `SubClassOf` or `EquivalentClasses` axioms. Indirect relation among tokens k' and k'' is given by:

$$OC(C_j) := \{C_k | \texttt{SubClassOf}(C_j,C_k)\} \cup \{C_k | \texttt{EquivalentClasses}(C_j,C_k)\} \quad (6)$$

$$IDR(k',k'') := \bigcup_{C_k \in OC(C_j)} match(L(C_k),k') \cup \bigcup_{C_k \in OC(C_j)} match(\texttt{name}(Px_{C_k}),k'') \quad (7)$$

where $OC(C_j)$ collects all other classes that are indirectly related to C_j through either `SubClassOf` or `EquivalentClasses` axioms, and $IDR(k',k'')$ is similarly constructed from classes in $OC(C_j)$.

The direct and indirect relations enable effective query interpretation during the search process. The classes matched via direct relations have the classname and its related context words in the query. The indirect relations are used to extract classes whose names may not directly appear in the query.

4.4 Search Results Ranking

The search results are ranked by the ranking model. The ranking model is built using *learning to rank* (LTR) algorithm [17, 16, 12], which are supervised machine learning algorithms designed to build ranking models. Training data is generated from querylogs. Feature vectors (FV's) were generated for each combination of query and result in the querylog. Components of such FV's are ranking features. The ranking features are the similarity value between query and all the context axioms and properties of concept. The training data of feature vectors was then used to build learning to rank (LTR) models, based on the RankLib implementation[1]. The ranking model learns the weights features using the ranking model. The models are then applied to get the scores for all the pairs of query and concept search results. These scores are sorted to obtain the top-k ranked search results.

5. EVALUATION

5.1 Benchmark dataset

We compare our system with the search function of BioPortal,[2] a large and widely-used biomedical ontology repository. In our experiment a large portion of ontologies, 296 openly available in total, were downloaded from BioPortal. Together these ontologies contain 2,062,080 classes. Our index maintains the term frequency of all terms as well as co-occurring tokens required for co-occurrence statistics. These frequencies are further used in calculation of lexical co-occurrence of terms in a query. These single and co-occurring tokens in queries found using Pearson's Chi square statistics are input to our query interpretation procedure. They are first used to find direct relations among the query tokens. We further use them to extract results of indirect relations, which are found via equivalent-class or subclass-of axioms.

5.2 Evaluation Measures

The real user queries from the BioPortal querylog (July 2012 to July 2014) were used to evaluate our framework. More than 50% queries are multiple-token queries. We use the 2,000+ queries to evaluate our search results. The standard IR ranking measures are used for evaluation vis-a-vis BioPortal performance. The BioPortal querylog consists of *query*, *clicked-position*, *clicked-ontology-id* and *concept-id*. We first selected the queries for which the click information was available. We then selected the queries for which the click ontologies are openly available so that evaluation could be fair. (Some of the ontologies such as MEDDRA are not openly available.) The final set of queries used for evaluation is 2,173. We used the standard IR measures Mean Reciprocal Rank (MRR) and Normalized distributive cumulative gain (NDCG) to compare our search results vis-a-vis BioPortal.

5.3 Result Analysis and Discussion

Our results indicate better performance for the multitoken queries when the query keywords are related to each other. Most users enter keywords that are related to each other. These query tokens have high co-occurrence values and/or they have a direct or indirect relation among them. This is being used by our approach to interpret the query and understand the intent of the user. The average NDCG and MRR for our approach are 0.72 and 0.60, and those of the BioPortal are 0.61 and 0.42 respectively for multitoken queries. We also evaluated the system for single-token queries. The average NDCG and MRR values for our approach are 0.63 and 0.51, and those of BioPortal are 0.62 and 0.49, respectively. Our results are found to be statistically significant with, 95%, confidence level using Wilcoxon signed-rank test. We outperform the BioPortal, especially for multitoken queries. We evaluate the difference between the NDCG value for each query. The difference is calculated between NDCG value using our approach and that using BioPortal for the same set of queries. Please refer to Figure 2 for a detailed analysis. The NDCG values for 1000+ queries (>50%)

[1] http://people.cs.umass.edu/~vdang/ranklib.html
[2] http://BioPortal.bioontology.org/

are better for our results depicting better overall performance. More than 700 queries (>35%) have same level of performance. The better performance is for the queries where the tokens in the query are related. This is detected using our co-occurrence procedure and direct and indirect relation based context-driven approach. The number of queries with negative difference is 300+ (>15%) where BioPortal performs better and may be attributed to the ontology popularity considered while ranking.

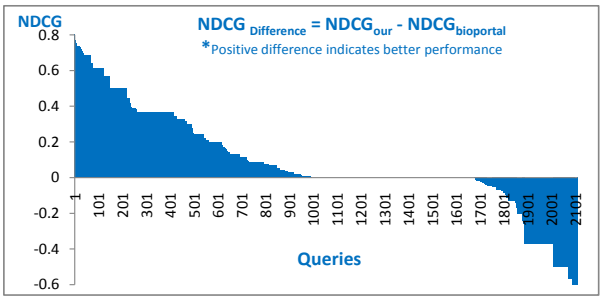

Figure 2: NDCG@10 Difference

6. CONCLUSION AND FUTURE WORK

Searching the right concept in a large ontology repository is a challenging task. In this paper, we present a novel concept search approach that incorporates three major techniques: (1) Context for each concept extracted by properties and axioms relevant to the concept, (2) Query interpretation based on the extracted context and (3) Ranking search results using *learning to rank* algorithms. Our comprehensive evaluation that involves more than 2,000 queries and 2,062,080 concepts shows that, for multi-token queries, our approach outperforms BioPortal's search function on two widely-used IR performance measures, NDCG and MRR. In future we plan to evaluate our search system using human-based evaluation. We will be comparing our performance with other related keyword based concept search systems in addition to Bioportal system. We also plan to evaluate our system with and without the learning to rank model. We will be designing more features such as ontology popularity to further improve the performance of our ranking model.

Acknowledgments

We thank Prof. Mark Musen and Paul Alexander of Stanford University for sharing the BioPortal querylog.

7. REFERENCES

[1] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *In Proc. of the International Semantic Web Conference, ISWC*, pages 5–9. Springer-Verlag, 2006.

[2] G. A. Atemezing and R. Troncy. Information content based ranking metric for linked open vocabularies. In *Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014, Leipzig, Germany, September 4-5, 2014*, pages 53–56, 2014.

[3] A. S. Butt, A. Haller, and L. Xie. Ontology search: An empirical evaluation. In *The Semantic Web–ISWC 2014*, pages 130–147. Springer, 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*, WWW '08, pages 1101–1102, New York, NY, USA, 2008. ACM.

[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, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *In Proceedings of the 4th International Semantic Web Conference*, pages 156–170, 2005.

[7] T. Finin, Y. Peng, R. Scott, C. Joel, S. A. Joshi, P. Reddivari, R. Pan, V. Doshi, and L. Ding. Swoogle: A search and metadata engine for the semantic web. In *In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 652–659. ACM Press, 2004.

[8] B. Florian and K. Martin. *Linked Open Data: The Essentials - A Quick Start Guide for Decision Makers*. edition mono/monochrom, Vienna, Austria, 2012.

[9] T. Franz, A. Schultz, S. Sizov, and S. Staab. *TripleRank: Ranking Semantic Web Data by Tensor Decomposition*. 2009.

[10] A. Hogan, A. Harth, and S. Decker. Reconrank: A scalable ranking method for semantic web data with context. In *In 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[11] 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. {JWS} special issue on Semantic Search.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of The ACM*, 46:604–632, 1999.

[14] I. Kollia, B. Glimm, and I. Horrocks. Sparql query answering over owl ontologies. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 382–396, Berlin, Heidelberg, 2011. Springer-Verlag.

[15] M. Konrath, T. Gottron, S. Staab, and A. Scherp. Schemex - efficient construction of a data catalogue by stream-based indexing of linked data. *J. Web Sem.*, 16, 2012.

[16] H. Li. A short introduction to learning to rank. *IEICE Transactions*, 94-D(10):1854–1862, 2011.

[17] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.

[18] N. F. Noy, P. R. Alexander, R. Harpaz, P. L. Whetzel, R. W. Fergerson, and M. A. Musen. Getting lucky in ontology search: A data-driven evaluation framework for ontology ranking. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, pages 444–459, 2013.

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

[20] Y. Qu and G. Cheng. Falcons concept search: A practical search engine for web ontologies. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41(4):810–816, 2011.

[21] U. Sawant and S. Chakrabarti. Learning joint query interpretation and response ranking. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1099–1110, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[22] T. Tran, H. Wang, S. Rudolph, and P. Cimiano. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 405–416, Washington, DC, USA, 2009. IEEE Computer Society.

[23] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In *ISWC/ASWC*, pages 552–565, 2007.