A Taxonomy of Semantic Web Data Retrieval Techniques

Anila Sahar Butt
Australian National University
Canberra, Australia
anila.butt@anu.edu.au

Armin Haller
Australian National University
Canberra, Australia
armin.haller@anu.edu.au

Lexing Xie
Australian National University
Canberra, Australia
lexing.xie@anu.edu.au

ABSTRACT

The Semantic Web provides access to an increasing amount of structured information in a wide variety of domains. Information overload due to the large amount of structured data is as much a problem as on the traditional Web. To solve this problem, ample research has been proposed on Semantic Web data retrieval techniques and after more than a decade of research in this domain it is now reasonable to consider the questions: is the field of Semantic Web data retrieval making progress? What are the directions that have been taken? and what are some of the promising significant directions to pursue future research? To answer these questions, we review the state-of-the-art Semantic Web data retrieval techniques and define a taxonomy of these techniques to classify the ongoing research and find potential future research directions.

Keywords

Taxonomy, Semantic Web search, Data retrieval approaches

1. INTRODUCTION

With the increasing popularity of the Semantic Web, there is a continuous growth in the amount of publicly available OWL and RDF(S) datasets on the Web. The problem of finding information in this huge amount of data is rapidly becoming as challenging a problem as information retrieval on the traditional Web. The problem is at least mitigated by the fact that meaningful and actionable information for a user query can theoretically be retrieved by exploiting the inherent nature of the Semantic Web data. However, in regards to the retrieval techniques, there is a wide range of work originating in different communities available that claims some sort of relevance to Semantic Web data retrieval. For example, terms encountered in the literature which claim to be relevant for Semantic Web data retrieval include ontology search, linked data retrieval, entity search, sub graph matching etc. Given this diversity, it is difficult to identify the problem areas and compare some of the solutions. Part of the issue is the lack of a comprehensive survey, a standard terminology, hidden assumptions or undisclosed technical details, and the dearth of evaluation metrics.

K-CAP 2015, October 07-10, 2015, Palisades, NY, USA.
 ISBN 978-1-4503-3849-3/15/10 ...$15.00
DOI: http://dx.doi.org/10.1145/2815833.2815846.

This paper aims to address some of these gaps. It contributes a survey of prominent historical and state-of-the-art techniques for Semantic Web data retrieval (SWR). We develop a taxonomy for SWR techniques consisting of 16 dimensions which are grouped into five topics: retrieval aspects, storage and search approaches, ranking, evaluation, and practical aspects. Each of these five topics consists of three or more dimensions. After discussing these 16 dimensions, we provide a brief review on how existing SWR techniques fit into our taxonomy. Based on our review, we are able to identify gaps in existing techniques, which allows us to highlight future research directions.

The rest of this paper is structured as follows. In Sec. 2 we present the general Semantic Web data retrieval process to provide the reader with the necessary background required to understand our taxonomy of SWR techniques. Sec. 3 describes the dimensions we identified that allow us to characterize SWR techniques, and we describe how some prominent SWR techniques fit into our taxonomy. We summarize the characteristics of all surveyed techniques in Table 1. We then discuss directions for future research in Sec. 4, and conclude the paper in Sec. 5.

2. SEMANTIC WEB DATA RETRIEVAL PROCESS

Data retrieval on the Web is a complex process consisting of several steps. A common Semantic Web data retrieval framework is similar to a typical Web search process as shown in Fig. 1. Here, boxes denote components of the data retrieval process and lines indicate data flow among these processes. Most of the processes are concerned with the pre-processing phase (i.e. data acquisition, warehousing, indexing and reasoning), while the remaining processes are concerned with the query-time processing phase i.e., query evaluation. Ranking may be part of the pre-processing or the query-time processing phase depending on the approach.

The first step, data acquisition (i.e. data crawling and parsing), is crucial for the retrieval approaches; because the quality of any retrieval system depends on the quality of the underlying dataset. Data acquisition necessitates Web crawlers, more specifically structured data crawlers for Semantic Web data crawling. The purpose of these crawlers is to gather a collection of linked data as quickly and effi-

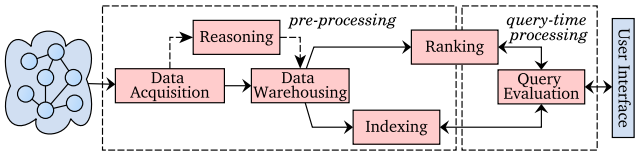

Figure 1: Outline of the general Semantic Web retrieval process

ciently as possible, while providing at least the required features for respecting the limitations imposed by publishers (i.e. politeness and robustness). A large number of linked data crawlers has been proposed including [3, 24, 29]. These crawlers mostly gather data from the Web by traversing the linked graph. Moreover, some crawlers also clean the data syntactically. The output of the data acquisition process is materialized for further processing.

The graph-based nature of RDF(S) data necessitates special data structures for data storage. The Semantic Web community has proposed a variety of storage structures as discussed in Sec. 3.2.2. Some SWR approaches also infer implicit data (triples) from the crawled data before materializing it. To infer these logical consequences from the set of asserted facts or axioms, special purpose reasoners are designed and reused by the community [18, 40]. Most of the research in this area (reasoning) is conducted separately, but not in the context or as a part of SWR approaches, therefore reasoning approaches are not covered in this work. However, there are existing benchmark studies that compare features and performance of the different available reasoners [4].

In large Semantic Web data collections, finding a match for a structured query or a keyword query requires lots of comparisons that are neither feasible nor necessary, because of infeasible query response times or the large number of non-matching triples in the data collection, respectively. Indexing techniques are required to mitigate this problem. A single word, a URI or a combination of URIs, commonly called the key, is used to decide where to find or insert data on disk. As with traditional Web information retrieval techniques, indexing has a trade-off between the computational complexity and the quality of the matching results. Having many small, but more specific keys in an index (more filtering) will result in a smaller candidate result set and thus reduce the computational cost, but at the same time it is more likely that some possible (partial) matches are being missed. On the other hand, a less specific key will result in a lager candidate result set but likely to more exact matches. Various techniques for indexing linked data have been developed; an analysis on indexing, based on the structure and the content of the key is presented in Sec. 3.2.3.

In addition to providing information in response to a user query in real time through indexing, some retrieval approaches also provide a ranking of the results. The ranking tries to determine which result is the most appropriate for the query. A substantial amount of ranking modes are designed or adopted for Semantic Web data ranking as discussed in 3.3. Some SWR techniques rank data in a data collection (i.e. corpus) off-line, independent of a user query, and materialize ranks along with indexing and others retrieve results for a query and apply ranking models to rank retrieved results only. Once the indexing and ranking is finished, the Semantic Web data becomes available for retrieval. Like Web search engines, SWR techniques allow users to explore linked data through keyword queries, but also through a structural query model where a user can pose additional more complex navigational queries. For this purpose, the user interface has to provide some means for a user to specify these complex queries. The users pose their queries through interfaces and the queries are mostly evaluated in a bottom-up fashion. i.e., the first match of the content of a resource is found with the help of the index and then the other information is used for filtering and result selection through real-time queries.

3. A TAXONOMY OF SEMANTIC WEB DATA RETRIEVAL TECHNIQUES

In this section we describe a taxonomy for SWR techniques. Our aim in developing this taxonomy is to provide a clearer picture of current approaches to retrieve Semantic Web data, and to identify gaps in these techniques which will help us to identify future research directions. We describe 16 dimensions of these techniques which we categorized into five main topics, as illustrated in Fig. 2. In the following sections we discuss each dimension in detail, while we also provide an overview of the methodologies or techniques applied in these dimensions.

3.1 Retrieval aspects

All existing SWR approaches can be categorized into three major dimensions with respect to the retrieval design decisions: the type(s) of the data that can be explored with the approach, the way(s) a user can initiate the retrieval process, and the type(s) of the output as a result of a user's query.

3.1.1 Retrieval Scope

SWR techniques can be classified into those that explore schemata defined by ontologies describing a conceptualization for a domain of interest and those that explore data generated according to these schemata. The former are referred as '`ontology-retrieval techniques`' and the latter as '`linked-data-retrieval techniques`'. The linked-data-retrieval techniques [22, 32, 44] focus on the retrieval of entities, relationships among entities, and sub-graphs. While the ontology-retrieval techniques [1, 15, 45, 7] find the classes and properties within or across ontologies, and ontologies themselves. Both these type of approaches focus on different components of the retrieval process. The large size of linked data available requires retrieval techniques to mainly focus on efficient indexing and query evaluation plans. On the contrary, datasets that only consist of ontologies are relatively small and thus the ranking of results is more relevant in the retrieval process than the indexing and efficient query plan execution. '`Graph-retrieval techniques`' [20, 48] is a category of SWR techniques comprised of the approaches proposed for general graph-based data but which are also applicable to and/or tested on the Semantic Web data retrieval task.

3.1.2 Query Model

SWR techniques generally consider one or more out of four query models: `keyword search`, `structured query search`, `faceted browsing`, and `hyperlink-based navigation`. In keyword-based SWR techniques [15, 20], a user poses a query

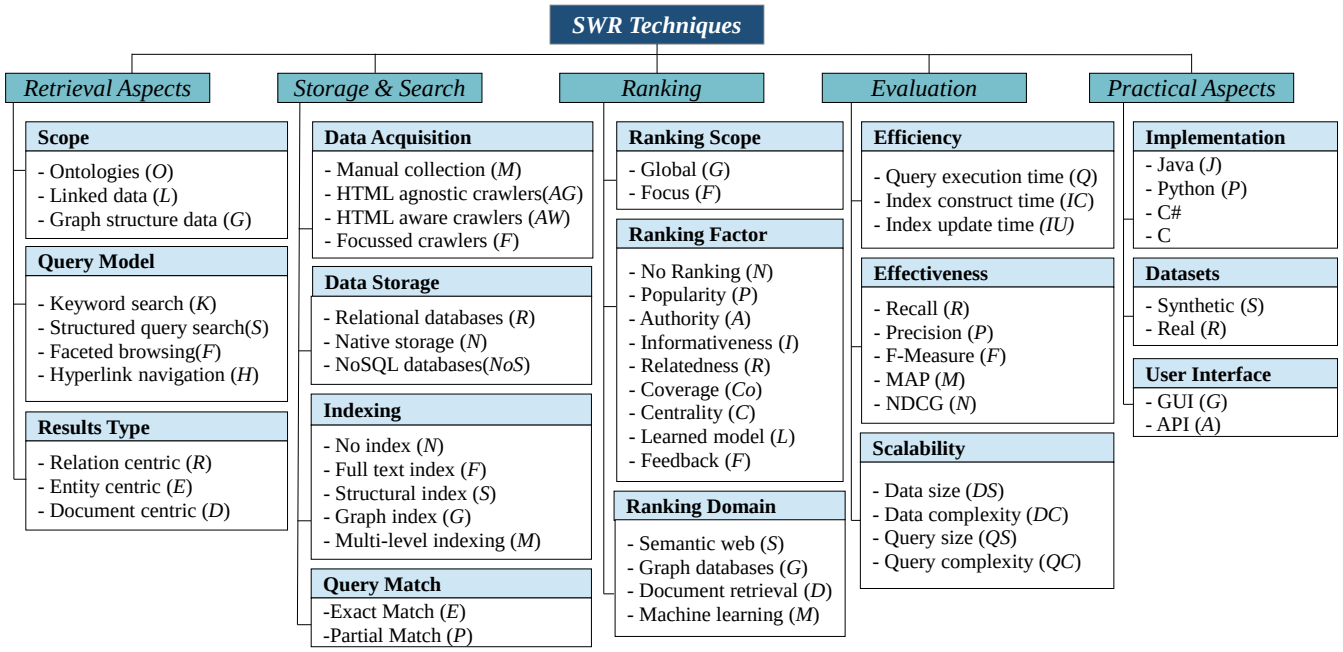

Figure 2: The dimensions used to characterize Semantic Web retrieval techniques

string composed of one or more keywords, while the results are retrieved based on a match to one or more keywords in the query string. Structured query search introduces complexity while providing more flexibility to meet the user's requirements by retrieving results for a user specified pattern. Most of the SWR techniques [13, 45, 46, 47, 49] provide simply an endpoint to query the data through the SPARQL structured query languages or graph queries. Few techniques [31, 33] allow the user to find and filter the results based on a faceted browsing approach. Each facet is a characteristic (i.e. property) and the facet values are object values for that characteristic. Facets can be fixed irrespective of the search result as defined by the UI developer, or generated dynamically based on the characteristics of the search results [33]. Hyperlink-based techniques [13, 41, 32] facilitate users to navigate within the data. Each hyperlink is a predefined query that is executed when a user clicks on it. Keyword-search, faceted browsing and hyperlink-based navigation facilitate naïve users in exploring data, whereas structured query interfaces are for expert users; they need to know the syntax of the query language and the underlying schema of the data.

3.1.3 Result Type

The examined SWR approaches mostly consider one of three different output types to facilitate users in the exploration of the data. (i) `Document-centric` approaches [32, 7] list URIs or labels of matched documents (i.e. ontologies) and/or document parts (i.e. classes, properties and entities). The document-centric approaches may list URI's of same ontologies (resp. resources) multiple times containing different pieces of information about ontologies (resp. resources). (ii) `Entity-centric` approaches consolidate available data about the entity from multiple documents and the consolidated information is presented as a profile of the entity [22, 44]. Therefore, rather than listing matched documents as incomplete pieces of information, an entity-centric search outputs one or more matched entities with their available profile in the dataset. (iii) `Relation-centric` approaches [2, 11] find relationships between entities. Mostly, structured queries or faceted browsing helps to perform relation-centric retrieval.

3.2 Storage and search approaches

3.2.1 Data Acquisition

The quality of a retrieval system depends on the quality of the underlying dataset. Data collection is mostly done in two ways: (1) *manual collection* – an admin or an owner collects a dataset manually considering the requirement or scope of the designed approach, and (2) *linked data crawler* – an application that gathers a collection of linked data as quickly and efficiently as possible. Existing linked data crawlers can be divided into three categories based on their crawling approach: (i) `HTML agnostic crawlers` do not crawl HTML documents. Therefore, these crawlers [23] are not able to discover linked data embedded in HTML documents and RDF documents surrounded by HTML documents. (ii) `HTML aware crawlers` crawl both RDF and HTML documents and follow RDF and HTML links within them. However, when crawling, the crawler visits many HTML documents that have no embedded linked data and do not point to any RDF documents. (iii) `Focused crawlers` use a limited HTML crawling approach in order to control the efficiency of HTML crawling. These crawlers crawl both RDF and HTML documents but limit the crawling space for HTML documents. For example, [32] crawls only those HTML documents that are explicitly provided as endpoints by users and extracts embedded linked data and 'href' links with '.rdf' extension within them.

3.2.2 Storage

SWR techniques generally consider one of three storage structures: (1) `Native Storage:` SWR approaches [22, 20] deploy persistent storage with their own designed storage architecture and are generally considered to be more efficient than the ones relying on relational databases [9]. (2) `NoSQL Databases:` Some of the SWR techniques use NoSQL databases to increase processing power and storage. Hadoopis one of the most widely used NoSQL databases, used for example in Sindice [32]. (3) `Relational Databases:` SWR approaches employ tradtional relational database management systems such as Microsoft SQL, MySQLto store triples or quads. Semantic Web data is stored in a *vertical representation* - a big triple table or quad table, or in a *horizontal representation* - property tables and vertical partitioning. This storage approach was mainly adopted by approaches [15] in the early days of the Semantic Web, but due to the slow response time its no longer a choice.

3.2.3 Indexing

Various techniques for indexing linked data have been developed since the advent of the Semantic Web; and several surveys of these techniques have been presented [27]. In this work we divide RDF data indexing into four major categories. The investigated SWR techniques implement one or more types of these four indexes.

Full-text Index: is implemented as an inverted index composed of a lexicon, i.e., a dictionary of terms that allows fast term lookup; and of a set of inverted lists, one inverted list per term. However, compared to traditional document-based inverted indexes, the difference is in the structure of the inverted lists. Based on the structural difference in the inverted list full-text indexes in Semantic Web are further divided into *node-based full-text indexes* and *graph-based full-text indexes*. In node-based indexes (resp. graph-based indexes) the inverted lists are composed of a list of the resource/node identifiers (resp. ontologies identifiers) for each terms of the lexicon. To improve the space and time complexity of full-text indexes, some SWR approaches separate node-based full-text indexes for entities, attributes and object values into an *entity-node inverted index*, *attribute-node inverted index*, and *value-node inverted index*.

Structural Indexes: are specially designed for RDF data stores [19]. Such indexes can be classified into those that index a triple (subject-predicate-object) and those that index a quadruple (context-subject-predicate-object). The former are known as *triple indexes* and the latter as *quad indexes*. In contrast to a separate index on subject, predicate, object and/or context where join operations are required to derive the answer for a query, a complete index on a quad or triple pattern allows a direct lookup on multiple dimensions without a join operation. To make the search more efficient, indexes with all possible patterns for a quadruple or a triple are implemented, i.e. 4^2 =16 and 3^2 =9 indexes for quadruple and triple respectively.

Graph Indexes: Recently, graph indexes have been introduced to support efficient structural queries over graph or RDF data. Compared to traditional indexes where each node has a key-value pair in the index, the difference is in the content structure of the key-value pair in the graph index. Traditionally, the key in an index node is either a text or an identifier; in graph indexes a key is a subgraph (patterns) and its value is a set of database graphs (ontologies) that contain the subgraph. Data structures adopted to implement graph indexes to enhance the filtering include *feature-matrices* [47], graphs [48, 46], and lattices [49].

Multi-level Indexes: Other than creating multiple type of indexes SWR techniques also introduce multi-level indexes to improve the efficiency of the retrieval process. One such approach is presented in [20]. Indexes at different levels narrow down the search space by reducing the size of relevant dataset to the query.

3.2.4 Query Match

The efficiency and effectiveness of the query evaluation is heavily influenced by how the matches are found in the data collection. The matched results for a query in a repository are found either for an `exact match` or for a `partial/approximate match`. The exact match is effective since an exact keyword or structure query match always ensures the right answer for a user; however, it sometimes results in an empty result-set if either an exact match is unavailable or the user is unaware of the contents or the structure of the dataset. On the other hand, a partial match enhances the chances to come up with approximate or similar results for the user, but with the disadvantage of a potentially large number of results that need some ordering mechanism to suggest the most appropriate result to the user.

3.3 Ranking

In addition to providing the information in response to a user query, some retrieval approaches rank the results. The ranking indicates which result (entity or ontology) is deemed to be the most appropriate for the query. The ranking models designed or adapted for Semantic Web data ranking can be distinguished along several dimensions; some of them are discussed in this section.

3.3.1 Ranking Scope

Ranking Scope denotes if a SWR technique is query dependent or not. The query dependent approaches are referred to as '`focussed-ranking`'- i.e. the ranking model is applied only on the result set and the relative order of each result in the result set is computed. The second class of ranking approaches which we refer to as '`global-ranking`' are implemented on the complete dataset (ontologies or linked data) irrespective of the query. Since the focussed-ranking approaches are applied only on a subset (results) of the dataset they lead to a higher efficiency in computing the ranks; however, the ranks calculated are not the global optimum. Global-ranking is more time consuming, but computes globally optimum ranking scores of query results.

3.3.2 Ranking Factor

One other important dimension of ranking is the 'ranking factor' based on which the ranks are calculated. The factors that have been used in different ranking approaches are explained here:

Popularity: Similar to the document retrieval domain most of the ranking techniques adopted for Semantic Web data order the output of a user query in-terms of the popularity of a result in a dataset. Different SWR techniques have adopted different popularity measure models, originally designed for information retrieval. PageRank [34] and TF-

IDF [37] are the most widely used popularity measures for Semantic Web data ranking.

Authority: Authority, a measure of *trustworthiness*, is another factor on the basis of which individual resources or documents (ontologies) are ranked. HITS [26], designed for informational retrieval, is used to compute the authority of the resources in [7]; and variations of HITS are also investigated in [21].

Informativeness: For Semantic Web data, informativeness is a measure of the degree of information carried by each resource that helps to identify it. Several SWR techniques [30, 10] adopted Shannon entropy [39] as an informativeness measure, according to which informativeness of a resource is the negative log of the probability of presence of the resource in a given dataset.

Relatedness: Relatedness is the similarity between features (property-value pair) of a resource. A resource is ranked higher if features of the resource are related to each other. Different relatedness models have been proposed such as WordNet to measure the relatedness between two features based on their text similarity; or *distributional relatedness* i.e., two features are more related if they more often co-occur in a certain graph (ontology).

Coverage: Coverage is a query-dependent ranking factor that measures how much of a query term or a structured query is covered by a resource. The Vector Space Model (VSM) [38] and BM25 [36] are document retrieval models that compare similarity between a query and the matched document. These models have been adapted on the task of resources and ontology ranking.

Learning a model: Other approaches for ranking Semantic Web data are rooted in 'learning to rank', a technique developed for machine learning [43]. In these approaches, different graph/ontology features are selected (or computed) and on the basis of these features a ranking model is learnt and then the learned model is used to produce the ranking for search results [8].

Centrality: Some ranking models designed or adopted by SWR approaches consider centrality of a concept/resource to compute their ranks. Some approaches find the centrality as connectivity of a node/resource in a graph/ontology [16]. Mostly, it is a measure of the number of relations or edges for a concept or a node.

User Feedback: Some SWR techniques [31] consider user feedback such as view count and query log to compute the ranking of the result-set.

3.3.3 Ranking Domain

SWR techniques are either designed purely for Semantic Web data or are borrowed from other domains. Most of the approaches are adopted from the '*document retrieval*' domain including: Pagerank, HIT, VSM, TF-IDF and BM25. Because of the graph structure of the RDF model, many of the SWR techniques adopt ranking approaches that were designed for *graphs* in general, i.e. shortest path [17] and centrality measure [16]. A recent trend for ranking Semantic Web data is the adaptation of learning-to-rank [43] approaches from the *machine learning* domain. However, these models are not applicable to the Semantic Web data in its original form, because of the nature of the data. Therefore variants of these models are implemented and some of them are studied in [6].

3.4 Evaluation

The performance of a SWR technique needs to be evaluated in terms of three factors: efficiency, effectiveness, and scalability. The efficiency of a SWR approach provides a measure of how fast the retrieval process is, while the effectiveness of the approach is measured by the accuracy of the retrieval model and quality of the retrieved results. Scalability measures the SWR technique for its capability to handle large scale datasets and complex queries.

3.4.1 Efficiency

Efficiency is evaluated using measures that are dependent upon resource utilization on the computing platform (i.e. memory consumption) or measures that are based on the time taken to retrieve the relevant results. Existing approaches evaluate the time taken on different processes of the retrieval process including: (1) *query evaluation time*, (2) *Index construction time*, and (3) *Index updation time.*

3.4.2 Effectiveness

One or more out of five popular metrics are used to evaluate the effectiveness of an SWR approach.

Recall: a fraction of *relevant* documents that are *retrieved* i.e.

$$\begin{aligned} Recall &= \frac{\#(relevant - results - retrieved)}{\#(relevant - results)} \\ &= \frac{retrieved}{relevant} \end{aligned}$$

Precision: a fraction of *retrieved* results that are *relevant*

$$\begin{aligned} Precision &= \frac{\#(relevant - results - retrieved)}{\#(retrieved - results)} \\ &= \frac{relevant}{retrieved} \end{aligned}$$

It is hard to determine the relevance and irrelevance of all results for queries resulting in a larger number of matched results, therefore mostly precision is determined for a cut off value i.e. for top-k results. Precision at k (P@k) for a k value is calculated as:

$$p@k = \frac{\#\ relevant\ results\ in\ top\ k\ results}{k}$$

F-Measures: F-Measure is a measure that trades off precision versus recall which is the weighted harmonic mean of precision and recall, i.e.,

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Mean Average Precision: The average precision for the query Q of a SWR technique 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. MAP is defined as the mean of AP over all queries run in an experiment and is calculated as:

Table 1: Categorization of Prominent Semantic Web Retrieval Techniques

	Search Aspect			Storage/Search				Ranking			Evaluation			Practical Aspects		
Techniques	Search Scope	Query Model	Result Type	Data Aquisition	Data Storage	Indexing	Query Match	Ranking Scope	Ranking Factor	Ranking Domain	Efficiency	Effectiveness	Scalability	Implementation	Datasets	User Interface
LOV [45]	O	K,S	D	M	N	F	E	G	P	D	-	-	-	J	R	G,A
BioPortal [31]	O	K,F	D	M	-	-	E	G	F	D	-	-	-	J	R	G,A
OntoSearch2 [42]	O	K,F	D	M	N	S	P	G	Co	-	Q	-	DS	-	R	G
OBO [41]	O	H	D	M	-	N	-	-	-	-	-	-	-	-	R	G
OntoSelect [5]	O	K,F	D	M,AG	-	-	-	-	-	D	-	-	-	-	R	G,A
Swoogle [15]	O,L	K	D	AG	R	F	P	G	P	D	Q	-	-	J	R	G
WATSON [13]	O,L	K,S	D	AG	N	S,F	E	-	N	D	-	-	-	J	R	G,A
OntoKhoj [35]	O	K	D	AG	R	F	-	G	P	D	-	F	-	-	R	G,A
AKTiveRank [1]	O	K	D	-	-	-	P	F	P,Co,C	G,D	-	P	-	-	-	-
Sindice [32]	L	K	D	F	NoS	F	P	F	Co	D	Q,C,U	-	DS	J	R	G,A
Sig.ma [44]	L	K	E	-	NoS	F	P	F	A,Co	D	Q	-	-	J	R	G,A
SWSE [22]	L	K	E	AG	N	S	P	F	A,P	D	Q,C,U	-	DS	J	R	G
SemRank [2]	L	-	R	M	-	-	E	F	-	M	-	-	-	-	S	-
LTR [12]	L	K	E	M	-	-	-	G	L	M	-	N	-	-	R	-
SLQ [48]	G	K	D	M	N	S,F	P	F	C	M	Q	P,M,N	DS	J	R,S	-
OSQ [46]	G	S	D	M	N	G	P	-	-	-	Q,C	P,M,N	DS	-	R,S	-
Lindex [49]	G	S	D	M	N,M	G	P	-	-	-	Q,C,U	-	DS	-	R,S	-
NeMa [25]	G	S	D	M	N	G	P	-	-	-	Q,C,U	P,R,F	DS	-	R	-
BLINK [20]	G	K	D	M	N	M	P	F	C	G	Q,C	-	DS	-	R	-
SSSGD [47]	G	S	D	M	N	G	P	-	-	-	Q	-	QS,DS	-	R,S	-

'-'represents undefined or unknown dimension for an approach

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

Normalize Discounted Cumulative Gain (NDCG): NDCG is a standard evaluation measure for ranking tasks with a 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.

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

The NDCG is computed by dividing 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.

3.4.3 Scalability

SWR techniques can be evaluated using measures that are dependent on the size (no. of triples) and the structural complexity of the dataset and the query, and/or on the flexibility of the approach. The former is referred to as *space scalability*, and the latter is referred to as *structural scalability*. Evaluations are conducted to compute resources utilization (including memory and time). For a scalable approach the resources utilization does not grow to intolerable levels as the size or complexity of the data set or query increases. The metrics for scalability are characterised as: (i) *data size*, (ii) *data complexity*, (iii) *query size* and (iv) *query complexity*.

3.5 Practical aspects

The final category covers practical aspects of SWR techniques, including the type of datasets used for implementation or experimental evaluations, how the solution was implemented, and if a proposed solution was developed with a specific application area in mind.

3.5.1 Implementation

This dimension specifies the implementation techniques that have been used to implement or to prototype a SWR technique in order to conduct its experimental evaluation. Some solutions proposed in the literature provide only theoretical proofs but they have not been evaluated experimentally, or no details about their implementation have been published.

3.5.2 Dataset

Experimental evaluation on one or ideally several datasets is important for the critical evaluation of a SWR technique. Due to the difficulty in obtaining real-world data that contain a large number of triples, synthetically generated datasets are commonly used.

3.5.3 User Interface

Most of the SWR approaches are developed primarily for

interactive browsing while additionally providing programmatic access to its content. For browsing *Graphical user interfaces* (mostly Web-forms) are used to make it a more interactive experience for the user, while programmatic access is made available through *Web services* that enable application developers to use the content of the SWR techniques in their application.

4. DISCUSSION AND RESEARCH DIRECTIONS

In this section, we analyze the surveyed SWR techniques as characterized in Table 1 with regard to the proposed taxonomy. This analysis highlights several areas of potential future research directions in SWR. Since the beginning of the development of techniques that aim to provide solutions for SWR, there is a clear path of progress, starting from early techniques that solve the problem of Semantic Web data retrieval for exact keywords document search using naïve approaches, moving on to entity search techniques that allow advanced faceted browsing. Still, there are some research gaps that can be focused on in the future.

Dynamic Faceted Browsing: Most of the ontology search engines and libraries either do not facilitate faceted browsing at all or filter results based on fixed facets for all searches (e.g. LOV [45] and BioPortal [31]). A more satisfying approach seems to lie in finding facets dynamically based on the matched results for a query. However, a major hurdle in identifying the dynamic facets is the syntactic diversity in describing the same property, for example, a `title` of a resource can be described as a `name`, a `title`, a `label` etc. in different vocabularies. A potential solution might be in clustering similar types of properties into a single group using machine learning and data mining techniques; and declaring the group of properties as a facet rather than having individual properties as facets.

Ontology Retrieval: Most of the ontology search systems retrieve ontological terms (concepts and relations) and some provide ontology search based on some keywords. The ontology search systems that retrieve matched ontologies for multi-keyword queries often returns ontologies that match to one of the query terms. However, they lack a criteria to find the relevant ontologies that cover most of the query terms or related concepts to these terms. BioPortal [31] provides an opportunity to find an ontology based on its text description, however, it is a domain dependent ontology library and does not deal with all type of ontologies. A general solution for ontology retrieval based on text descriptions or several keywords still needs to be devised.

Ontology Ranking Models: Ontology collections are limited in size, therefore ranking becomes the core task for ontology search engines and libraries, rather than efficient search. However, ontology ranking is pragmatic, because search results are a match of a search term with a more expressive class, property or ontology description. There may exist many ontologies that contain concepts and relations with their labels matching the keyword query, however, they have been described differently mainly in terms of their: (i) `perspective` - A concept may be defined in different perspectives e.g., a person class is defined in many ontologies, for example, the 'foaf' 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's information need. Most of the ontology retrieval systems do not focus on ranking at all [14] and others adopted ranking approaches that are rooted in graph or document retrieval ranking models without considering the underlying nature of ontologies. This provides ample opportunities for research to significantly improve the ranking of ontologies or ontological terms based on a more expressive user query.

Linked data retrieval effectiveness vs. efficiency: linked data retrieval approaches can be classified into two major categories: (i) `Effectiveness oriented techniques` - which apply ranking models to retrieve the most appropriate answers [28, 21, 32] (ii) `Efficiency oriented techniques` - which mainly focus on efficient indexing to achieve the efficiency in retrieving results with less focus on ranking [20, 46, 48]. There is scope for linked data retrieval techniques that make a reasonable trade off between effectiveness and efficiency of the retrieval approaches.

Ranking of triples for entity retrieval: In recent years the linked data retrieval paradigm is shifting from document retrieval to entity retrieval [22, 32]. The entity retrieval process finds entities and consolidates attributes for an entity from multiple data sources. It requires a ranking of triples for the entity to prioritize relevant attributes of that entity. Existing approaches rank properties in a general context based on their occurrence in a dataset. However, the ranking of a property depends upon the entity it belongs to. The property may be attached to more than one entity and the relative importance of the property will vary for each entity. Secondly, the object values for multivalued properties mostly have different ranking criteria depending upon the entity to which the property belongs to, but they are also ranked according to its popularity in current approaches. This constitutes a significant gap between the state-of-the-art entity ranking techniques and the ideal ranking and presents opportunities for future research.

An evaluation framework for Semantic Web data retrieval techniques: There is currently no comprehensive evaluation strategy that facilitates the comparative evaluation of different SWR techniques with regards to their effectiveness, efficiency and scalability. Researchers have used a variety of evaluation measures and datasets (both real and synthetic), which makes comparing existing techniques difficult. It is currently not possible to determine which technique(s) perform better than others on data with different characteristics and of different sizes. So far, it seems that no single SWR technique has outperformed all other techniques in all aspects on large datasets.A benchmark on ontology ranking [6] has been published recently. It contributes an ontology collection, ten benchmark queries, a gold standard and evaluation of eight state-of-art ranking model on the task of ontology search. However, the benchmarks deals with ontology concepts ranking only. There is no comprehensive study that compares many existing techniques within the same framework and on different datasets.

Conducting such large experimental studies is one avenue of research that would be highly beneficial to better understand the characteristics of these techniques.

5. CONCLUSION

In this paper we have presented a brief overview of historical and current state-of-the-art techniques for Semantic Web data retrieval. We have identified 16 dimensions that allowed us to characterize these techniques, and to generate a taxonomy of such techniques. This proposed taxonomy can be used as a comparison and analysis tool for Semantic Web data retrieval techniques. Through this taxonomy we identified various shortcomings of current approaches that suggest several future research directions in this field.

6. REFERENCES

[1] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *The Semantic Web-ISWC 2006*, pages 1–15. Springer, 2006.

[2] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *Proc. of the 14th international conference on World Wide Web*, pages 117–127. ACM, 2005.

[3] A. Batzios, C. Dimou, A. L. Symeonidis, and P. A. Mitkas. Biocrawler: An intelligent crawler for the semantic web. *Expert Systems with Applications*, 35(1):524–530, 2008.

[4] J. Bock, P. Haase, Q. Ji, and R. Volz. Benchmarking owl reasoners. In *Proc. of the ARea2008 Workshop, Tenerife, Spain*, 2008.

[5] P. Buitelaar, T. Eigner, and T. Declerck. Ontoselect: A dynamic ontology library with support for ontology selection. In *The Semantic Web-ISWC 2004*. Citeseer, 2004.

[6] A. S. Butt, A. Haller, and L. Xie. Ontology search: An empirical evaluation. In *The Semantic Web–ISWC 2014*, pages 130–147. Springer, 2014.

[7] 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. Springer, 2014.

[8] A. S. Butt, A. Haller, and L. Xie. DWRank: Learning Concept Ranking for Ontology Search. In *Semantic Web Journal*. IOS, 2015.

[9] A. S. Butt and S. Khan. Scalability and performance evaluation of semantic web databases. *Arabian Journal for Science and Engineering*, 39(3):1805–1823, 2014.

[10] G. Cheng, T. Tran, and Y. Qu. Relin: Relatedness and informativeness-based centrality for entity summarization. In *The Semantic Web–ISWC 2011*, pages 114–129. Springer, 2011.

[11] G. Cheng, Y. Zhang, and Y. Qu. Explass: Exploring associations between entities via top-k ontological patterns and facets. In *The Semantic Web–ISWC 2014*, pages 422–437. Springer, 2014.

[12] L. Dali, B. Fortuna, T. T. Duc, and D. Mladenić. Query-independent learning to rank for rdf entity search. In *The Semantic Web: Research and Applications*, pages 484–498. Springer, 2012.

[13] M. d'Aquin and E. Motta. Watson, more than a semantic web search engine. *Semantic Web*, 2(1):55–63, 2011.

[14] M. d'Aquin and N. F. Noy. Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:96–111, 2012.

[15] 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 *Proc. of the thirteenth ACM-CIKM*, pages 652–659. ACM, 2004.

[16] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[17] A. V. Goldberg and C. Harrelson. Computing the shortest path: A search meets graph theory. In *Proc. of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 156–165. Society for Industrial and Applied Mathematics, 2005.

[18] V. Haarslev and R. Möller. Racer: An owl reasoning agent for the semantic web. In *Proc. of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with*, pages 91–95, 2003.

[19] A. Harth, J. Umbrich, A. Hogan, and S. Decker. Yars2: a federated repository for querying graph structured data from the web. In *Proc. of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 211–224. Springer-Verlag, 2007.

[20] H. He, H. Wang, J. Yang, and P. S. Yu. Blinks: Ranked keyword searches on graphs. In *Proc. of the 2007 ACM SIGMOD international conference on Management of data*, pages 305–316. ACM, 2007.

[21] A. Hogan, A. Harth, and S. Decker. Reconrank: A scalable ranking method for semantic web data with context. 2006.

[22] 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.

[23] 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.

[24] R. Isele, J. Umbrich, C. Bizer, and A. Harth. Ldspider: An open-source crawling framework for the web of linked data. In *The Semantic Web-ISWC 2010*. Citeseer, 2010.

[25] A. Khan, Y. Wu, C. C. Aggarwal, and X. Yan. Nema: Fast graph search with label similarity. *Proc. of the VLDB Endowment*, 6(3):181–192, 2013.

[26] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[27] Y. Luo, F. Picalausa, G. H. Fletcher, J. Hidders, and S. Vansummeren. Storing and indexing massive rdf datasets. In *Semantic Search over the Web*, pages 31–60. Springer, 2012.

[28] A. Maedche, S. Staab, N. Stojanovic, R. Studer, and

Y. Sure. Semantic portal-the seal approach. *Spinning the Semantic Web*, pages 317–359, 2003.

[29] R. Meusel, P. Mika, and R. Blanco. Focused crawling for structured data. In *Proc. of the 23rd ACM-CIKM*, pages 1039–1048. ACM, 2014.

[30] R. Meymandpour and J. G. Davis. Linked data informativeness. In *Web Technologies and Applications*, pages 629–637. Springer, 2013.

[31] 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, et al. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37:W170–W173, 2009.

[32] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice. com: A document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3(1):37–52, 2008.

[33] E. Oren, R. Delbru, and S. Decker. Extending faceted navigation for rdf data. In *The Semantic Web-ISWC 2006*, pages 559–572. Springer, 2006.

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

[35] C. Patel, K. Supekar, Y. Lee, and E. Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *Proc. of the 5th ACM international workshop on Web information and data management*, pages 58–61. ACM, 2003.

[36] 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.

[37] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[38] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[39] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[40] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[41] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, 2007.

[42] E. Thomas, J. Z. Pan, and D. Sleeman. Ontosearch2: Searching ontologies semantically. In *Proc. of the OWLED 2007 Workshop on OWL: Experiences and Directions*, volume 258 of *CEUR Workshop*, 2007.

[43] A. Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.

[44] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig. ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355–364, 2010.

[45] P.-Y. Vandenbussche and B. Vatant. Linked Open Vocabularies. *ERCIM news*, 96:21–22, 2014.

[46] Y. Wu, S. Yang, and X. Yan. Ontology-based subgraph querying. In *Proc. of 29th International Conference on Data Engineering*, pages 697–708. IEEE, 2013.

[47] X. Yan, P. S. Yu, and J. Han. Substructure similarity search in graph databases. In *Proc. of the 2005 ACM SIGMOD international conference on Management of data*, pages 766–777. ACM, 2005.

[48] S. Yang, Y. Wu, H. Sun, and X. Yan. Schemaless and structureless graph querying. *Proc. of the VLDB Endowment*, 7(7), 2014.

[49] D. Yuan and P. Mitra. Lindex: a lattice-based index for graph databases. *The VLDB Journal*, 22(2):229–252, 2013.