Inf Retrieval (2014) 17:295–325
DOI 10.1007/s10791-014-9240-0

A survey of approaches for ranking on the web of data

Antonio J. Roa-Valverde · Miguel-Angel Sicilia

Received: 22 January 2013 / Accepted: 20 March 2014 / Published online: 4 April 2014

Abstract Ranking information resources is a task that usually happens within more complex workflows and that typically occurs in any form of information retrieval, being commonly implemented by Web search engines. By filtering and rating data, ranking strategies guide the navigation of users when exploring large volumes of information items. There exist a considerable number of ranking algorithms that follow different approaches focusing on different aspects of the complex nature of the problem, and reflecting the variety of strategies that are possible to apply. With the growth of the web of linked data, a new problem space for ranking algorithms has emerged, as the nature of the information items to be ranked is very different from the case of Web pages. As a consequence, existing ranking algorithms have been adapted to the case of Linked Data and some specific strategies have started to be proposed and implemented. Researchers and organizations deploying Linked Data solutions thus require an understanding of the applicability, characteristics and state of evaluation of ranking strategies and algorithms as applied to Linked Data. We present a classification that formalizes and contextualizes under a common terminology the problem of ranking Linked Data. In addition, an analysis and contrast of the similarities, differences and applicability of the different approaches is provided. We aim this work to be useful when comparing different approaches to ranking Linked Data and when implementing new algorithms.

Keywords Linked data · Information retrieval · Semantic search · Ranking algorithms · Link analysis · Semantic web data management

A. J. Roa-Valverde (✉) · M.-A. Sicilia
STI Innsbruck, Innsbruck, Austria
e-mail: antonio.roa@sti2.at

M.-A. Sicilia
e-mail: msicilia@uah.es

1 Introduction

Scenarios characterized by searching and browsing on large volumes of data or documents require of special treatment in order to guide the users to the most relevant pieces of information. Typically, users have to select and filter all the information they go through until they find a relevant piece of data that matches what they are looking for. Also, user behavior studies have found out that users in Web search engines are viewing fewer result pages (Jansen and Spink 2006), which evidences the importance of ranking outcomes.

In the traditional Web the information space is modeled as a corpus of documents that establish links among them as an implicit way to state relationships within the information they contain. Users can make use of these links to navigate the information moving from one document to another using Web browsers. Following this model, referred to as the Web of documents, search engines were proposed as a way to facilitate the navigation towards finding the required information, and retrieval mechanisms have been devised that make use of known properties of the link structure (Broder et al. 2000), being a notable example the PageRank algorithm (Brin and Page 1998). Despite current Web document retrieval solutions have demonstrated to be useful, new challenges appear when dealing with finer-grained information spaces where entities formally described and the relationships among them play the main role, and not the documents where they appear or are mentioned (Sheth et al. 2004). These challenges are identified and described in detail in Sect. 2.

New methods for exploiting semantic relationships between data must be considered in order to make the most out of the information usage. These ideas are used and applied in the context of what is called the "Web of Data", described in Bizer et al. (2009) as "a Web of things in the world", in contrast to the traditional Web of documents mentioned above. Basically, what favors the trend from the Web of documents to the Web of Data relies on the limitations of human capabilities for consuming huge amounts of information and the need for data. This, together with the improvements on machine's power, helps to process the information and convert it into data ready for direct consumption. Furthermore, converting Web documents (unstructured data) to data (structured) helps to achieve data and service integration purposes. In what follows we describe the main elements of the Linked Data initiative[1] as it can be considered the cornerstone of the Web of Data nowadays.

In the last decade, methodologies from database, artificial intelligence, information retrieval and linguistics research have been combined under the idea of pursuing a Semantic Web that helps to overcome the challenge of dealing with vast amounts of heterogeneous information (Lassila 2007). All the efforts carried out to find a solution to this problem have produced different formalisms to model the knowledge implicitly contained in the information. Notably, the specification of the Resource Description Framework (Klyne and Carroll 2004), RDF Schema (McGuinness and van Harmelen 2004) and the Web Ontology Language (Brickley and Guha 2014) have been devised as languages for the representation of semantics. While having the required tools and capabilities to express the available knowledge, the fact of unifying all different perceptions of the real world under the same formal representation is still nowadays a challenge, due to the distributed nature of the Web that requires reconciling the semantics of disparate, heterogeneous schemas and representations. In order to overcome this problem approaches like the Linked Open Data initiative have arisen. As stated in Bizer et al. (2009) "Linked Data is simply about using the Web to create typed links between data from different sources".

[1] http://linkeddata.org/.

In this way, the Web of Linked Data aims at building a dynamic set of data modeled using very simple principles while still keeping a common representation of the shared knowledge. As outlined in Berners-Lee (2006), the main principles of Linked Data are:

1. Use URIs as names for things;
2. Use HTTP URIs so that people can look up those names;
3. When someone looks up an URI, provide useful information, using the standards (RDF, SPARQL);
4. Include links to other URIs, so that they can discover more things.

In addition, the main tasks that have to be performed in order to publish data as Linked Data are (1) to assign consistent URIs to data published, (2) to generate links, and (3) to publish metadata that allows further exploration and discovery of relevant datasets.

The Linked Data initiative has an enormous potential because it facilitates access to the very large amounts of information available on the Web in a structured and integrated fashion (Bizer et al. 2009). However, exploiting vast amounts of information requires new techniques that facilitate the user requirements for consuming and managing data. When searching for information, the fact of retrieving a significant collection of results satisfying the user requirements is very important, but the manner how these results are presented, filtered or ranked to the user can impact in a more important grade the way a user identifies the piece of information that better approximates to the target of his/her search. To help in this task ranking algorithms are used.

In a few words, a ranking algorithm implements a function that accepts a set of items and returns an ordered version of the set without modifying the items themselves. The function is implemented taking into account certain preferences that determine the order of the items. In this way, the same collection of items could be ranked following different approaches, i.e. different order functions. Whilst the area of information retrieval has addressed and provided different approaches for this problem, e.g. PageRank (Brin and Page 1998), HITS (Kleinberg 1998) and SALSA (Lempel and Moran 2001), there is still a lack of consensus referring to the problem of ranking structured data as that exposed in the Web of Linked Data. As stated previously, when dealing with structured information, entities and the relationships among them play the main role, and not the documents where they appear.

The motivation of this work is to formalize the problem of ranking linked data and give a comprehensive overview of existent ranking methods for the Web of Data. There are other survey studies concerning to the topic of semantic Web search (Hildebrand et al. 2007), where ranking algorithms for structured data are to some extent described. However, to the best of our knowledge none of the existing works gives a complete overview of ranking methodologies for the "Web of Data" that helps to understand the benefits and drawbacks of each one. This is of great importance for the future of the Web of Linked Data, as the same problems of volume that appear in the Web will arise as the Web of Data grows. The main target of this work focuses on helping researchers in the Semantic Web community to identify and understand the problem of ranking information. After a review of the literature, we have selected the most relevant algorithms according to their impact in this field. In this way, we have tried to homogenize the vocabulary employed with the aim of settling a common reference for semantic ranking methodologies.

In Sect. 2 we define and describe the challenges that appear when ranking the Web of Data. Section 3 defines the rank operator and shows a generic architecture for implementing it. Section 4 shows a possible classification for the different approaches. In sect. 5 we use the taxonomy proposed in 4 to discuss the different ranking techniques we have

identified as relevant. Section 6 provides an overview of the current evaluation approaches and the research directions in this field. Finally, in Sect. 7 we conclude this survey.

2 Open challenges

Along this survey we can appreciate several aspects related to the way ranking approaches are designed. These aspects are imposed by the new information needs of the Web of Data and can be summarized as follows:

1. Dealing with larger and heterogeneous information.
2. How to integrate both structured and unstructured information.
3. Query execution.
4. Consolidation of results.

In the following, we discuss each of these points.

2.1 Data heterogeneity

In the Web of Data, the underlying data structure is a directed graph containing the relationships (edges) between the different entities (nodes). The manipulation of this graph defines the behavior of each algorithm and so its efficiency and use for different goals. The granularity of the handled data determines the amount of information to consider during the ranking. Obviously, the more information the bigger the underlying graph will be, and so, more computation will be required to calculate the ranking scores. This issue has little flexibility in terms of choosing a specific approach, since it fully depends on the requirements of the final application. In this way, if the client application consumes information at entity level of granularity, the underlying algorithm necessarily will have to deal with relationships among entities. As stated in Coffman and Weaver (2010), a solution to tackle this problem must focus on developing better indexing techniques to produce systems with better scalability.

Heterogeneity in the Web of Data refers to the presence of information of diverse nature and contexts that is interlinked constituting a unique and global dataset. This fact is uncommon in traditional databases, where the existence of relational models and the application of normalization techniques limit the contextual dimension of the data. The price for having this flexibility in the Web of Data is that ranking approaches must develop powerful techniques to determine the relevance of the information and filter the content to suit the user expectations.

2.2 Data integration

This problem can be seen as a consequence of the heterogeneity discussed in the point 1. Authors in Balog and Neumayer (2013) and specially in Halpin et al. (2010) manifest that there is still a lot to do in order to exploit the information available in the structure of the Web of Data: “We found that keyword queries were taken as such, and despite our expectations they were not interpreted or enhanced with any kind of annotations or structures. The possibilities for query interpretation using background knowledge (such as ontologies and large knowledge bases) or the data itself is another characteristic of semantic search that will need to be explored in the future.” As we will see later in Sect. 6, this tendency of treating structured data as just documents was used in the initial attempts

for ranking the Web of Data. Current approaches rely on entities as the atomic part of the information to build the junction between structured and unstructured data. The main problem here is how to identify a given entity in the unstructured data.

2.3 Query execution

The dependency of the algorithm regarding to an external query impacts its time of response. In this way, when following a dynamic ranking strategy different queries can produce different sizes of the underlying graphs to analyze. Here the main drawback is narrowing the time of response, what makes this kind of algorithms not adequate for environments requiring almost real time responses. On the other hand, while the time of response is stable in static ranking, the main inconvenient of this methodology is its weakness against changes in data. Any update of the underlying graph implies the recalculation of all ranking scores.

Further effort needs to be put on developing powerful keyword query solvers that help to match the user query with the graph structure.

2.4 Consolidation of results

Unlike document oriented ranking, most of the new approaches dealing with ranking on the Web of Data return a combination of data that is present in different snippets. It is necessary to explore possible solutions to combine data that is relevant to the user while at the same time minimizing the amount of noise. A possible direction could be applying diversity ranking like in Zhu et al. (2007). Another possibility could be trying to develop aggregation functions within the ranking process like in Sawant and Chakrabarti (2013).

2.5 Additional problems

In addition to the previous problems, we identify some minor areas that need further research and development.

- How to add support for data streams and big data. The development of high scalable models is a must to deal with this kind of environments.
- Lack of mechanisms to capture the social dimension of linked open data and its exploitation in ranking. As described in Halpin et al. (2010), the wisdom of the crowd is starting to be considered in ranking tasks. Certainly in that concrete work authors use crowdsourcing platforms to evaluate the ranking assessments.
- How to include support for personalized ranking functions. If users are the consumers of the information they should be able to customize the consumption. The ranking factor is the cornerstone that defines the algorithm itself and its choice will determine the relevance of results. Usually each ranking approach makes use of one factor, implementing a fixed schema that imposes a single ordering on results. This means that the same query made for different purposes will always return the same ordering. In Anyanwu et al. (2005) authors state that some flexibility should be built into the relevance models so that different orderings may be imposed on the same result set depending on the user need. From this we can conclude that user data requirements and consumption needs guide the choice and preference of a specific algorithm over another.

3 The ranking problem

A ranking algorithm implements a function, which accepts a set of items and returns an ordered version of the set without modifying the items. The function is implemented taking into account certain preferences that determine the order of the items. In this way, the same collection of items could be ranked following different approaches, i.e., different order functions. Formally, a ranking algorithm implements a function of total order $f : X \rightarrow \Re$ such that for any items $a, b \in X : f(a) \leq f(b) \leftrightarrow a \prec b$, where $\prec$ defines a binary relationship on the set X. Note that $\prec$ makes reference to the factor that guides the ranking strategy. In the following, we will refer to this functionality as the Rank operator.

Every ranking algorithm implicitly implements a strategy of relevance. Relevance is a concept that symbolizes the grade of match considered between the retrieved information and the intention of the user. To clarify this, let us consider the example used by authors in Manning and Schütze (2008): "If a user types python into a web search engine, they might be wanting to know where they can purchase a pet python or they might be wanting information on the programming language Python". While it is difficult for a system to know what the information need is, the user can judge the returned results on the basis of their relevance to it. In order to improve the evaluation of the system, ranking algorithms aim to approximate the relevance as perceived by the user considering different features that are available in the data. From the user point of view, the relevance model implemented discriminates one ranking strategy from each other.

Architecture for generic rank

Considering a reference architecture helps to review and compare different ranking approaches. With this aim, in the following we describe a high-level architecture for a generic and customizable implementation of Rank.

Figure 1 depicts the components that gather the functionality of a ranking system. The clients are applications from diverse domains like search engines, data browsers, interlinking engines. Each client uses the generic implementation of Rank to determine ranking scores that are needed to support further domain dependent tasks.

The input to the generic implementation of Rank is raw data that needs to be inspected as a previous step before computing the ranking scores. The preprocessing module is in charge of isolating the ranking mechanism from any kind of additional complexity due to the heterogeneous representations of data. Without lost of generality, we assume that the outcome of this step is modeled as a graph containing the relationships among the different data items, regardless of the granularity (documents, datasets, entities, etc.).

The generic implementation of Rank computes a set of scores that are used to rank the items available in the input graph. These scores follow an internal representation that is tightly dependent on the implementation. An exporter exists that is able to translate the scores from the internal representation into the representation required by each client application.

In general, it is possible to obtain different ranking scores for a given data input. Therefore, the implementation of Rank should capture the requirements of the user to obtain high relevant item-score associations.

From the application perspective, the described architecture is independent of the underlying data model, which means that it can be applied to unstructured and structured data in a transparent way. After describing the different ranking solutions, the reader will note that the key factor that differentiates one approach from another relies on the way the

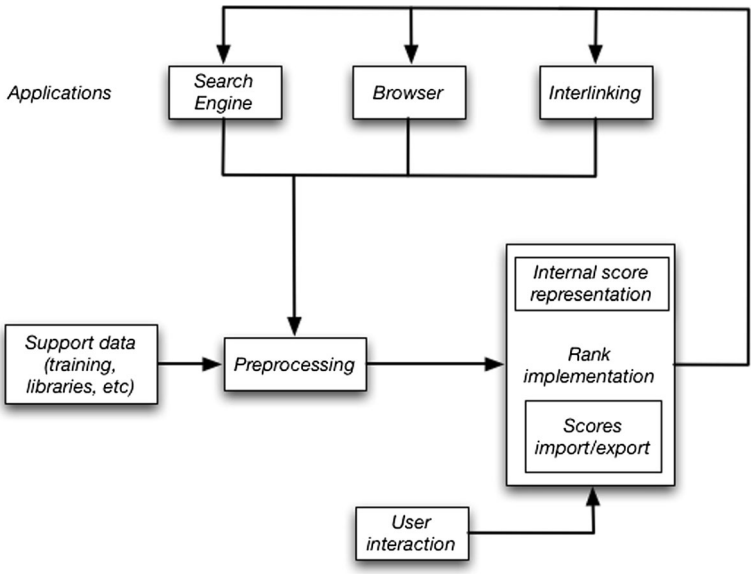

Fig. 1 Generic architecture for rank

preprocessing component is instantiated. When ranking Web of Data, the preprocessing implementation and the modeling of its output are determined by how the issues described in Sect. 2 are addressed. At this stage, we can state that the main difference between ranking the Web of documents and ranking the Web of Data remains on how the input data is handled and which of the features available within the data are exploited by the Rank implementation.

4 Classification of structured data ranking approaches

In this section we classify the major contributions for ranking structured data. Figure 2 depicts the classification scheme we propose.

An implementation of Rank can rely on multiple ranking algorithms or *rankers*. This allows the user to choose the ranker that better fits her relevance model. Within this setting there are two possible configurations. First, there is the realization of individual rankers, each of which computes its ranking scores based on a single relevance criterion. Second, there is the combination of individual rankers, either by using multiple ranking criteria within an integrated *hybrid ranker* or by combining multiple ranking scores produced by different ranking algorithms within a *composite ranker*. A further refinement in this dimension is whether the combination is carried out manually or automatically. For individual rankers we identify the following dimensions:

- *Queries* Ranking approaches can consider user queries for computing the scores.
- *Features* A Feature is an aspect which exists in the data that makes possible to establish a comparison between items belonging to a data source.
- *Granularity* It refers to the granularity of the data source to be ranked (i.e., entity, identifier, relationships, dataset, document).
- *Heuristics* A heuristic makes reference to a specific mechanism that guide the score computation process.

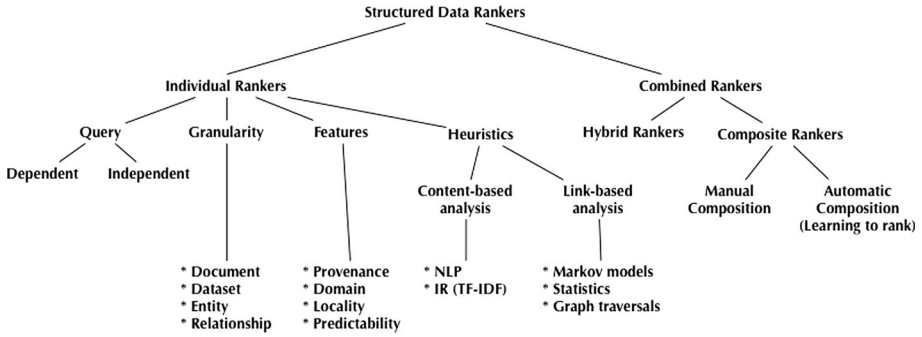

Fig. 2 Classification of ranking approaches

In the following sections we discuss the individual approaches and focus on the hybrid rankers as they are the most common implementations in the literature. We leave the composite rankers out of the scope of this survey. We are aware that composite rankers, and specifically, learning to rank (LTR) has got a lot of attention in Information Retrieval due to its probed efficiency and effectivity (Liu 2009). LTR approaches apply machine learning in order to induce a ranking model from given training data. The main focus of LTR techniques is to tune the learning algorithm, which is completely independent of the underlying data structures. This fact makes of LTR applicable for both structured and unstructured data without needed customization. The reader can observe this in works like Dali et al. (2012) and Chen and Prasanna (2012), where authors apply LTR to RDF rankers. For an extensive compilation on LTR for Information Retrieval, we refer the reader (Liu 2009).[2]

4.1 Query dependency

Query dependency makes reference to the way in which the user input is considered during the ranking. By definition a ranking algorithm is query dependent (also called dynamic ranking) if the function implemented ranks the set of items regarding to the user input. This mechanism does not use previous results or the implicit structure of the dataset, but the ranking process is calculated on the fly. On the opposite, a query independent algorithm (also known as static or absolute ranking) relies on the internal structure of the dataset to rank the items, regardless of the user query. It is possible to find algorithms that combine both static and dynamic strategies within the same implementation as in Balmin et al. (2004) and Anyanwu et al. (2005).

4.2 Granularity

When ranking information, it is important to decide the nature of the items to rank as well as the level of detail that will determine the order of results after the ranking. We distinguish between item and ranking granularity. Item granularity refers to "the granularity of what is being ranked", i.e., the nature of the items, e.g., a document or an individual RDF resource. Ranking granularity makes reference to "how the information represented

[2] http://research.microsoft.com/en-us/um/beijing/projects/letor/paper.aspx.

by the items is processed", i.e., the aspects taken into account to compute the ranking. In this way, an algorithm could compute the ranking of all RDF resources ("what") considering the relationships established among them and the relevance of documents where they appear ("how").

Item granularity establishes an abstraction of the relationships between the items in such a way that defines a scale that ranges from fine to coarse grained. In general, the granularity determines somehow the amount of information represented by each item. The following scale covers the range of granularities that can be found nowadays on the Web:

- *Dataset* A dataset is a collection of data with several characteristics that define its structure and properties. An example of dataset is DBpedia[3] and Freebase.[4]
- *Document*. A document is a bounded artifact that represents certain information. Note that different documents could belong to the same dataset.
- *Entity*. As defined in Delbru et al. (2010) an entity is "a self-contained unit of information that has relationships with other entities". Typical examples of entities could be persons, places or events. Through this work we refer to entity and resource as synonyms.
- *Identifier*. The idea of having a unique identifier per entity that helps to differentiate one thing from each other is supported by the Linked Data practitioners. However, guaranteeing the uniqueness of identifiers in a distributed environment like the Web is a difficult challenge that often produces the existence of several identifiers for the same entity. The ranking process could be simplified if two identifiers are interpreted as referencing to the same entity.
- *Relationship*. Intended as the predicate of a triple in RDF notation.

4.3 Features

A feature is an aspect which exists in the data that makes possible to establish a comparison between elements belonging to a dataset. The exploitation of distinct features produce different ranking results over the same data. We are aware of the fact that usually different works use different terminology for referring to the same concept. As this could be a problem to fully understand the main differences among ranking approaches, we have tried to unify them under the same terminology. The following summarizes the main factors used in previous works on ranking.

Provenance It makes reference to the origin of the information. This concept relies on the authority of a data source to guess how reliable the information is. The fundamental behind this idea is that the information is more trustable when it comes from a known source. In some works like (Cyganiak et al. 2008), authors use the term context as synonym of provenance: "(...) context typically denotes the provenance of a given statement". Here, authors introduce the idea of a N-Quad as an extension to the N-Triple notation for RDF, whose main aim is keeping intact the original provenance of each statement when publishing RDF data. The difference of the N-Quad notation regarding to the N-Triple is the introduction of a forth element which is the HTTP URI from which the statement was originally retrieved. Other authors refer to provenance as authority. Nevertheless, in this survey we will consolidate the terms provenance and authority uniquely as provenance, while using the term context as synonym of domain or topic of the data.

[3] http://dbpedia.org/.

[4] http://www.freebase.com/.

Domain It refers to the topic addressed by the data. For example, data about drugs in the context of Biology or data about artists in the context of music. Note that in this survey we will use domain and context as the same thing. On the other hand we will not use the term context as synonym of provenance as stated previously.

Locality When dealing with different data sources, locality refers to the concrete source that contains the data that we are considering. This idea goes further than the provenance of information, adding more details in situations where the same authority provides data distributed in different datasets. While the idea of provenance only takes into account who is the creator of the information, the conceptual location is supported through locality.

Predictability This concept was first introduced in Anyanwu et al. (2005) as a way to consider the gain of information in the ranking processes. The main idea is that data should not be considered to produce the same degree of relevance in every situation. For example, when dealing with two different domains, the same item could get a higher score for one domain than for the other. The measure that determines when the same data item is likely to produce different scores is precisely the predictability. Predictability is closely related to the concept of entropy in information theory.

4.4 Heuristics

The function implemented by a ranking algorithm is primarily determined by which data analyses are performed and how they are combined.

4.4.1 Content-based analysis

Content analysis involves those techniques that drive the extraction of features from the content ignoring any kind of relationships available in the data. The basis of content analysis techniques builds on top of string comparisons and parsing theories. The application of this kind of analysis to linked data also follows these ideas, but it requires a preprocessing phase that extracts the targeted text from the RDF graph. In RDF most of the text can be found as literals associated to widely used properties as *rdfs:label*. Depending on the schema and the vocabulary used to model the data more free text can be found. For instance, in the case of DBpedia we can find the property *dbpprop:abstract*, which contains text information about the abstract of the respective resource of Wikipedia that is referenced.

The goal of studying the content is that of establishing a set of measures relying on terms appearing in the text that helps to correlate the grade of similarity among different resources, therefore defining a function of ranking. Within this category we can find different groups of techniques, e.g. text mining and information retrieval. Text mining focuses on deriving information from text. As an example of text mining we can find named entity recognition (NER), which consists on identifying sequences of text referring to entities like countries, people, dates, etc. As we will see later in the discussion, some algorithms like Mirizzi et al. (2010) rely on resource similarity to compute the global ranking. Here, the similarity is obtained taking into account associated text labels to resources. Authors check for instance if the *rdfs:label* associated to a resource A is contained in the *dbpprop:abstract* of a resource B and vice versa. In Alani et al. (2006) authors determine the coverage of an ontology for the given search terms. For doing so, the proposed algorithm compares the terms within the query with class labels. In Anyanwu et al. (2005) authors proposed what they call S-Match, used for determining the degree of similarity of a user keyword and a property occurring in a semantic association.

Complementary to text mining, also information retrieval techniques can be found in some ranking algorithms. Traditionally, the problem of ranking information has been studied in the area of information retrieval and therefore the use of this kind of techniques appears straightforward. Information retrieval ranking strategies rely on statistical models that measure the existence of relevant terms in documents. Note that documents are intended as the granularity of items to be ranked. In this way, documents are considered as containers of textual information, which are independent from each other (links inside the documents, if any, are not taken into consideration). Relying on the measures derived from the content, the relevance of documents is ranked towards the user queries. DING (Delbru et al. 2010) is an example of ranking approach using a variant of the well-known TF-IDF measure applied to linked data, among other techniques.

Applying content analysis to data is not an accurate task, mostly due to the different interpretations of the information. As an example, let us consider the word "Cordoba", which can make reference to a city in the South of Spain and at the same time to a city in Argentina. Still more dramatic is the word "Java", which could refer to the programming language or to the Indonesian island. Previous works facing this problem have been published in Roa-Valverde et al. (2011) and Dbpedia spotlight (2011). In order to overcome ambiguity issues, content analysis is in many cases combined with other approaches like link analysis. The idea behind the combination is to extract the context around resources to improve the disambiguation of information.

4.4.2 Link-based analysis

With the arrival of the Web, link analysis was proposed as a new method to rank the hypertext documents considering the way the information is represented and related (Getoor and Diehl 2005). Unlike content-based ranking, link analysis strategies try to incorporate structural features of the information during the ranking computation. Link analysis is a technique relying on examining the graph structure established among items, where the nodes of the graph are the items to rank and the edges are the relationships or links among items. By inspecting the graph structure implicit properties can be derived and included in the ranking process. Link analysis can be thought as a case of success, which was originally implemented in algorithms like PageRank (Brin and Page 1998), HITS (Kleinberg 1998) and SALSA (Lempel and Moran 2001), commercially exploited by the most popular search engines. Inspired by this philosophy several extensions have been developed that increase the corpus of link analysis methodologies,[5] namely:

- *Weighted link analysis* The aim of this technique is to assign more relevance to certain kind of links depending on its type during the ranking computation. A major challenge is how to assign the weight to the links without having negative performance implications. Most of the approaches under this classification were proposed in the topic of database research, and for this reason they are not directly applicable on web-scale. As an example for this category we can consider the works described in Xing and Ghorbani (2004) and Baeza-Yates and Davis (2004).
- *Hierarchical link analysis* This technique performs a layered exploration of the underlying data and it is intended for distributed environments. For example, first considering relationships among super nodes or datasets and secondly considering

[5] This classification has been taken from Delbru et al. (2010).

relationships among resources. An example for this category can be found in Xue et al. (2005).

- *Semantic Web link analysis* This family of methods tries to exploit the semantic of relationships during the ranking process. This technique can be thought as an evolution of the weighted link analysis applied to the Semantic Web context. As an example we can consider the algorithms described in Finin et al. (2004) and Anyanwu et al. (2005).

5 Sample approaches from the literature

In the following, we group the algorithms in two main clusters, i.e., first the approaches designed for a semantic search context and second the approaches targeting keyword search in databases. While the second group is not dealing with RDF and open data, we still consider keyword search relevant because of the similar problems it tries to solve. Nevertheless, the main focus of this paper are the approaches within the first group. For more information about the basics of keyword search in databases we refer the reader to other surveys like Yu et al. (2010).

In Table 1 we show how a total of 16 ranking approaches (12 semantic search approaches and four keyword search approaches) fit the classification introduced in Fig. 2. The table focuses on the query dependency, granularity and ranking features, providing a comparison of the different techniques. The table shows that all approaches include different ranking criteria. In the case of keyword search approaches we can observe that all the implementations present the same characteristics, which manifests the nature of a more limited problem in comparison to semantic search. Most of the semantic search approaches implement static ranking and converge in targeting entities and their relationships as the ranking granularity. In general, most of the exploited features fall into provenance and context.

5.1 Semantic search

In this section, we discuss the different algorithms for ranking information on the Web of data. The reader can notice a refinement in the analysis and data features considered by the

Table 1 Summary of features for the surveyed ranking approaches

			OntologyRank	PopRank	SemRank	ReConRank	AKTiveRank	NAGA	Hart et. al	TripleRank	RareRank	DBpediaRanker	DING	Tonon et al.	BANKS	DISCOVER	ObjectRank	BLINKS
	Query dependency	Static ranking	x	x	x			x	x	x	x	x	x	x	x	x	x	x
		Dynamic ranking			x	x	x								x	x	x	x
Granularity	Item granularity	Document	x				x											
		Dataset							x				x					
		Entity(Resource)		x		x		x		x	x	x	x	x	x	x	x	x
		Identifier							x									
		Relationship			x			x		x				x	x	x	x	x
	Ranking granularity	Document relationships	x															
		Entity (Resource) relationships		x	x	x		x		x	x		x	x	x	x	x	x
	Features	Provenance				x		x	x	x	x		x		x	x	x	x
		Context (Topic)		x	x					x	x	x	x	x			x	
		Locality											x					
		Predictability			x			x										

different approaches that remarks the transition from heavy ontology documents to specific type of entities and their relationships.

5.1.1 OntologyRank

This algorithm was first introduced in Finin et al. (2004) as the strategy followed by the Swoogle semantic Web search engine to measure the importance of a semantic Web document (SWD). In this work, a SWD is considered as a Web document containing information which is modeled as an ontology or part of it, regardless of the underlying format used for its representation. SWDs have the same treatment than traditional Web documents, where the information is partitioned in smaller fragments, which are the documents themselves. Specifically, authors define a SWD as follows: "We define a Semantic Web Document (SWD) to be a document in a semantic web language that is online and accessible to web users and software agents. Similar to a document in information retrieval (IR), a SWD is an atomic information exchange object in the Semantic Web".

Authors distinguish between two kinds of documents, which they refer to as Semantic Web Ontologies (SWOs) and Semantic Web DataBases (SWDBs). These make reference to the definitions of T-Box and A-Box in the description logic literature, respectively. Since a SWD may consists of both T-Box and A-Box, authors consider a document to be a SWO when a significant portion of the statements it makes define new terms or extend terms defined in other SWDs by adding new properties or constraints. On the other hand, a document is considered as a SWDB when it introduces individuals and makes assertions about them or makes assertions about individuals defined in other SWDs, without adding or extending any term.

Following a link analysis approach similar to commercial search engines, Swoogle, through the use of OntologyRank, calculates the relevance of SWDs taking into account the following kind of relationships among documents:

- *imports(A,B)* A imports all content of B;
- *uses-term(A,B)* A uses some of terms defined by B without importing B;
- *extends(A,B)* A extends the definition of terms defined by B;
- *asserts(A,B)* A makes assertions about the individuals defined by B.

Swoogle relying on the random surfer model introduced by PageRank proposes an extension that accounts for the various types of links that can exist between SWDs. With this extension authors aim to uniform the probability of following a particular outgoing link. For doing so, OntologyRank assigns different weights to the four categories of inter-SWD relations.

This method is to be applied in scenarios characterized by the existence of ontology documents, where the main aim is to carry out some kind of analysis independently of the user's information needs. The advantage of this method in this kind of settings relies on its speed for computing the ranking scores due to the limited size of the required data model. The controlled size of the data model can be explained because of the limited amount of relationships that are examined during the ranking process. Another characteristic of this algorithm is its simplicity, as it does not make use of any ranking feature like provenance or context analysis. This fact can be an advantage or a limitation depending on the usage.

A use case of this algorithm can be found in Sicilia et al. (2012).

5.1.2 PopRank

PopRank (Nie 2005) is a PageRank-based algorithm that takes into account the semantic of the relationships among different objects within a specific domain (context in Table 1). Authors describe a case study where they rank the objects within a collection of authors, conferences and journals. The method considers both the Web popularity of an object, in terms of its input and output linkage, and the object relationships to calculate its popularity score. The authors define a Web object as a piece of information within a Web document. However, in order to maintain the same nomenclature with the rest of algorithms in this survey, we can consider a Web object as a resource, without loss of generality.

PopRank extends PageRank by adding different weights, namely popularity propagation factor (PPF), to each link depending on the type of relationship. With the aim of facilitating the assignment of weights, the authors propose a data mining mechanism to automatically add this weights to the links. "The simulated annealing algorithm is used to explore the search space of all possible combinations of propagation factors and to iteratively reduce the difference between the partial ranking from the domain experts and that from the learned model". As can be observed in this statement, the authors rely on the use of ranking lists, previously made by domain experts, with the aim of training the system.

Analyzing the whole graph is expensive in terms of performance because of the size of the search space. To face this problem, only a part of the graph is taken into account during the learning process without loss of accuracy, since the objective is not getting the exact ranking scores but the relative rank of the training objects.

Authors state in their experiments an average accuracy increment of 50 % over a baseline PageRank. They justify this improvement in the way the different weights are assigned. Finally, authors conclude that due its context independency, the algorithm can be applied to other search domains like eCommerce.

5.1.3 SemRank

In contrast to other approaches mostly centered around ranking entities, the work described in Anyanwu et al. (2005) proposes the alternative of performing ranking of relationships between resources. Authors argue that in the context of ranking relationships information retrieval techniques are not applicable as there is no way to determine how good a query matches a relationship like in the case of documents or entities. To overcome this problem, authors propose the use of information theory techniques relying on how predictable a result might be for users. This approach is referred to as discovery search, in contrast to the conventional search based on information retrieval.

An important feature of SemRank is that users can change the ordering of results depending on their need, overcoming the limitation of most ranking approaches which have a fixed ranking scheme that determines a unique ordering on results. In this way, and depending on the domain of the application, SemRank allows the user with the possibility of modulating the search by means of varying the mode from a conventional search mode to a discovery search mode.

The final score computed by SemRank is based on a combination of several factors, namely, the predictability or gain of information of a semantic association, the degree of similarity of a keyword and a property occurring in a semantic association (named S-Match by authors), and the refractive count of an association or the amount of differences between the properties that compose a path and the properties defined in the original schema.

Authors evaluated SemRank on a synthetic dataset modeling relationships among objects from four different domains, i.e., University, Banking, Flight and Organization. Unfortunately, they do not provide information about possible improvements over other baseline algorithms.

5.1.4 ReConRank

ReConRank (Hogan et al. 2006) is a combination of two different algorithms: a first one called ResourceRank, with the aim of ranking RDF resources and a second one called ContextRank, which tries to improve the ranking quality by introducing provenance information. ReConRank considers that resources are nodes in the graph (subjects and objects in triple notation). In the same way, ReConRank defines a context as the provenance or source of data, i.e., the algorithm measures the reliability and suitability of the data used for describing a resource in order to quantify the trustworthiness.

In spite of being a PageRank-based algorithm, there is an important difference in the ranking strategy, that is, ReConRank only analyses result data that matches the query introduced by the user performing a dynamic ranking. Authors argue that the static strategy followed by PageRank is not feasible when dealing with RDF data because the entire dataset might be excessively large and it would require data structures that exceed the amount of memory available. Also, updating the RDF data would suppose the ranking to be recomputed.

Another inconvenient of performing a static ranking is that scores are not correlated with user queries, which means that returned results might not be the most relevant for the user preferences. ReConRank computes the ranking of result data matching a user query. This result data is a topical subgraph containing those resources matching the query as well as neighbor resources that can be reached after n hops in the graph. This parameter determines how broad or narrow a search result is.

There are two main challenges that have been addressed by the authors. The first one is how the topical subgraph can be extracted from the RDF graph considering the existence of poorly interlinked datasets. The second challenge is to make the ranking algorithm fast enough for ranking during query-time.

ResourceRank This part of the algorithm is responsible for ranking resources in the RDF graph. In the same way than PageRank, this method follows an iterative computation over a connectivity matrix which is derived from the graph structure. Before the first iteration, PageRank initializes all nodes with an equal score. This is the first difference with ResourceRank, which assigns to each node the ratio of all links that it receives as inlinks. Authors state that using this approach reduces the number of iterations required by about one third. Another difference in ResourceRank with regard to PageRank is the inclusion of weightings during the computation, which can be manipulated in order to tune the ranking function as required by the user. Further details about the application of weightings functionality can be found in Hogan et al. (2006).

ContextRank ContextRank is just an extension to ResourceRank with the aim of including provenance during the ranking computation. Note that authors use the term context as synonym of provenance. On the Web, everybody can contribute information about certain resource by reusing its URI. This fact, that produces a strong community driven knowledge base, has the disadvantage of creating data that might be unreliable for describing a resource. ContextRank exploits this aspect in RDF data helping to quantify trust on different data sources.

In the following, we describe how the provenance information is extracted from the RDF graph for later inclusion in the ranking process.

The graph containing only the contexts is extracted from the original RDF graph. Note that contexts might also be resources. It might be the case that the derived graph is not well interlinked. However, in such situation, the graph can be populated with implicit relationships:

- Link between contexts and resources included on it.
- Link between resources and containing contexts.
- Link from a context to a context. When a resource in context A links to context B, a link from A to B is implied.

The result of applying these implicit relationships is a graph that combines resources and contexts. In order to isolate the rankeable entities, only those resources and contexts appearing as subject at least once in a triple are considered. This graph is used as input for the ranking computation.

Authors state that in spite of calculating two different ranking scores, i.e. one for resources and one for contexts, these results have been calculated relying on a unique graph and so they are dependent on each other. The goodness of ReConRank relies on the relationship between resources and its provenance. Computing ranking on the unified graph produces better results than ranking resources and provenance individually. According to the description of the algorithm, ReConRank does not combine both scores using any kind of weighting in order to unify them, this responsibility is left to the final consumer, e.g. the user interface. In this way, authors propose a possible serialization of results taking into account both ranking measures by using different colors depending on the provenance.

5.1.5 AKTiveRank

AKTiveRank (Alani et al. 2006) is an algorithm to rank ontologies relying on their relevance to a given query. The ranking mechanism is based on estimations about how well the different ontologies represent the terms of the query. The different metrics are only applied to ontology classes, ignoring instances and properties. It uses Swoogle (Finin et al. 2004) to get the list of ontologies to be ranked. Once the ranking is performed, the results are returned to the user as an OWL file containing the URIs of the different ontologies together with their total ranks. Relying on a number of structural metrics four measures are calculated independently for each ontology. The resulting values will be combined to obtain the total rank for the ontology.

- *Class match measure (CMM)* This measure evaluate the coverage of an ontology for the given search terms. AKTiveRank compares the terms of the query with class labels, scoring higher those ontologies which contains exactly the terms. It also considers partial matches, which are obviously regarded as worse than exact matches.
- *Density measure (DEM)* DEM is intended to approximate the representational-density or information-content of classes and consequently the level of knowledge detail. This may include how well the concept is further described (number of subclasses), number of properties associated with that concept, number of siblings, etc. Instances are not considered to equalize the treatment of populated and non-populated ontologies.
- *Semantic similarity measure (SSM)* This measure calculate how close are the concepts of interest relying on the ontology structure. SSM is measured from the minimum number of links that connects a pair of concepts.

- *Betweenness measure (BEM)* This measure calculates the number of shortest paths that pass through each node in the graph. Nodes that occur on many shortest paths between other nodes have higher betweenness value than others. AKTiveRank assumes that if a class has a high betweenness value in an ontology then this class is graphically central to that ontology. The ontologies where those classes are more central will receive a higher BEM value.

The global score for each ontology is calculated as a weighted combination of the previous measures.

In the same way than OntologyRank (Finin et al. 2004), AKTiveRank relies only on structural metrics available on the ontology documents. This makes both algorithms comparable regarding to its usage. Regarding its accuracy, authors state an accuracy of 90 % when compared against a gold standard.

5.1.6 YAGO-NAGA

In Kasneci et al. (2008) authors describe the NAGA semantic search engine (Not Another Google Answer). As described in Suchanek et al. (2007), NAGA constructs a knowledge graph that is used for answering user queries. As some queries can return multiple answers, NAGA implements a ranking mechanism based on generative language models (Liu and Croft 2005). The ranking model exploits the notions of confidence, informativeness and compactness. Confidence means how trustable the results are. For its computation authors use the provenance of information through a PageRank-like algorithm.

Informativeness refers to the amount of information represented by certain result. Following the same example used by the authors, in a query about Albert Einstein, results including information about his career as physicist should rank higher than results about politics, because Einstein is well known as a physicist rather than a politician.

Compactness refers to the structure of the graph to rank the results, i.e., direct connections are preferred to loose connections between entities. The compactness of answers is implicitly captured by their likelihood given the query. This is because the likelihood of an answer graph is the product over the probabilities of its component facts. Therefore, the more facts in an answer graph the lower its likelihood and thus its compactness. NAGA's granularity are facts. A fact can be a simple RDF statement or a complex graph structure containing multiple RDF statements.

Confidence and informativeness are two complementary components of the NAGA model. The confidence expresses how certain we are about a specific fact, independently of the query and of how popular the fact is on the Web. The informativeness captures how useful the fact is for a given query. This depends also on how visible the fact is on the Web. In this line, the definition of informativeness differs from the information theory one, which would consider less frequent facts as more informative. Note that this is the case in algorithms like Anyanwu et al. (2005). NAGA uses an analogy to TF-IDF, which gives more relevance to less frequent facts from the knowledge base.

5.1.7 Hart et al.

Harth et al. (2009) propose an algorithm to rank structured data like RDF. The authors remark the fact that in heterogeneous environments like the Web, the large number of data sources exhibits an enormous variability in the vocabularies used. In order to overcome this issue, they state that "the authority of data sources is an important signal that the ranking

algorithm has to take into account". The authors define the concept of naming authority, previously introduced in Kleinberg (1998), as "the data source with the power to define identifiers of a certain structure". The naming authority can be seen as the provenance of a piece of information.

Using a PageRank-based algorithm, the authors compute an absolute ranking for the whole RDF graph, regardless of a particular query. The algorithm determines rankings for sources and identifiers included in those sources, taking into account the provenance of the information through the naming authorities.

Authors evaluate their algorithm using two different RDF datatsets crawled from the Web. They state that regarding the quality, their algorithm gives better results than a PageRank baseline. In addition authors justify that "...we did not compare to ObjectRank because ObjectRank requires manual assignment of weights to each of the thousands of properties in the dataset, which is infeasible."

5.1.8 TripleRank

TripleRank (Franz et al. 2009) is a HITS-based algorithm (Kleinberg 1998) that exploits the RDF graphs with respect to three different criteria, namely the relevance of resources, relevance of objects and relevance of properties. The authors use the term authority as synonym of in-link degree, however it is important to clarify that other algorithms consider the authority like synonym of provenance. The relevance of objects is referred to as hub score by the authors. This measure is determined by the out-link degree. The relevance of properties is referred to as latent topics of interest or contextualization.[6] This measure takes into account the semantic of links during the ranking. For computing the scores, TripleRank relies on a 3-dimensional tensor to represent each one of the criteria. The output of the algorithm is a set of rated RDF resources, objects and properties making reference to hub (navigational) issues, authority and topic, respectively.

The algorithm consists of the three following steps:

1. Data collection and transformation. TripleRank requests RDF data from the linked open data cloud performing a breadth first exploration of the surrounding resources to a starting set of URIs. The exploration is limited by depth, number of statements and number of links to follow for each resource and link type. The collected data is then transformed to its tensor representation.
2. Pre-processing. The reason of this step is twofold. First to reduce the amount of data to be analyzed. Second, to increase the quality of the collected data. For doing so, the authors have consider the following strategy. On the first hand, predicates linking the majority of resources are pruned because they convey little information and dominate the data set. In this way, a threshold of 40 % has been defined, so that, predicates that occur in more than 40 % of all statements are pruned. On the second hand, to avoid the negative effect of dominating predicates the dataset has to be weighted. Weights are chosen so that all predicates are treated with the same grade of importance.
3. Analysis. In this step the PARAFAC[7] decomposition of the tensor is performed.

[6] Note this differs from the intended notion of context agreed in this survey.

[7] http://www.models.life.ku.dk/rasmus/presentations/parafac_tutorial/paraf.htm.

5.1.9 RareRank

Wei (2009) introduces the idea of the Rational Research model to emulate the search behavior of a "rational" researcher in a scientific research environment and propose the RareRank algorithm for ranking entities in semantic search systems.

RareRank is a modification of the PageRank algorithm where the random component (loosely known as "random surfer") has been replaced for a more deterministic component. The reason for this change is that in the specific domain of research there is less randomness than in Web search.

The computation of ranking scores combines the link information (e.g., a citation between two publications), and the content information (e.g., provided by the links between document-topic and topic-topic). While link information makes reference to physical links between resources, content information refers to related information to the existent information in resources for which there are not previously modeled links. The latter is modeled through the use of an ontology, which allows the navigation from a document to another through the previously non existing links, in addition to the citation links. Citation links are modeled as a knowledge base containing the relationships between entities. The model provides an appropriate basis for ranking various types of entities and clearly can be generalized into other domains.

To compute the ranking, RareRank relies on the same mathematical model than PageRank. Both use a Markov chain defined by a stochastic transition probability matrix, however there is a difference in the way in which RareRank defines the transition matrix. RareRank uses two transition graphs: the ontology schema graph and the knowledge base graph. The ontology schema graph contains the relations between domain classes and their transition weights. The knowledge base graph consists of instantiations of classes and their relationships from the ontology schema. The weight of a relationship between two instances i_a and i_b is determined as a combination of the following factors: the weight of the relationship between the classes of i_a and i_b as defined in the schema graph, the amount of instances of the same type than i_b pointed by i_a, and the strength of the association between the instances.

The ranking vector is computed in the same way than PageRank, i.e., applying the power iteration method to the transition probability matrix. The initial values of the matrix can be set up to $1/N$. After some iterations, the probability values start to converge to the invariant distribution, which corresponds to the ranking scores.

5.1.10 DBpediaRanker

In Mirizzi et al. (2010) the authors have proposed an algorithm called DBpediaRanker to rank resources on DBpedia with the aim of generating ad-hoc tag clouds regarding to a given query. The notion of domain context is introduced to reduce the search space and improve the search results. A domain defines a set of nodes within the same category in DBpedia, e.g., the IT domain or the tourism domain.

Given a set of feed nodes (previously identified by a domain expert) representing the context of search, the algorithm explores the DBpedia graph trying to retrieve all the nodes within the same domain. The exploration is carried out dereferencing links between nodes in the graph. During the exploration, the algorithm calculates the similarity between pairs of nodes within the domain. The core functionality of the algorithm relies on the way in which the similarity is implemented.

Given two resources r_1 and r_2, their similarity value evaluates the importance of the relationship between them. To compute this similarity, DBpediaRanker combines the information within the RDF graph with external information such as the output of search engines and social tagging systems. In a first stage, the algorithm verifies how many web pages include the value of the *rdfs:label* associated to r_1 and r_2, respectively. Then it compares these values with the number of web pages including both labels.

In a second stage, the algorithm exploits further information from DBpedia using the property *dbpprop:wikilinks*. This property represents a hypertext link between two documents in Wikipedia. When the algorithm finds this relation from to or vice versa, it assumes a strong relation between both resources.

Moreover, the algorithm checks if the *rdfs:label* of r_1 is contained in the *dbpprop:abstract* of r_2 and vice versa.

Finally, the similarity value is calculated as the sum of the others measures. The result is a contextualized weighted graph where the nodes are DBpedia resources and the weights represent the similarity value between two nodes. The similarity is the value that determines the relative relevance of the different resources with respect to the user query.

5.1.11 DING

DING or Dataset rankING (Delbru et al. 2010) is a PageRank adaption to the Web of data that aims to exploit locality of entities by following a hierarchical approach in two layers. An entity is a self-defined unit of information that maintains relationships with other entities. Entities and their relationships are grouped in datasets. The union of datasets and their relationships built the aforementioned Web of data. As defined in Alexander et al., "a dataset is a collection of data, published and maintained by a single provider, available as RDF, and accessible, for example, through dereferenceable HTTP URIs or a SPARQL endpoint".

DING uses links between datasets and combines the resulting values with semantic-dependent entity ranking strategies. Relying on the links established among different datasets, DING constructs a graph that is used for performing a coarse link analysis during the ranking process. A fine-grained link analysis is performed considering links among entities within each dataset. Both intra and inter-dataset links are contemplated.

As a general overview, DING performs ranking in three main steps: (1) dataset ranking, (2) entity ranking within the same dataset and (3) global entity ranking by the combination of both (1) and (2).

At the top level, DING computes linkset weights to consider the relevance of links between two different datasets. The aim of these weights is to estimate the probability of a user to go from a dataset to another following certain kind of link. In a similar way to the TF-IDF measure in information retrieval, authors propose the Link Frequency—Inverse Dataset Frequency (LF-IDF) measure to estimate the value of these weights. This measure takes into account both the number of links contained in a linkset and the general importance of the label involved in the link. The way how the linkset weights are calculated gives more relevance to a link with a high frequency in a certain dataset and low dataset frequency in the dataset collection. This means that links with labels whose authority is owned by a specific dataset are to have more importance than links with labels commonly used. For example, *dbpprop:reference* links defined by dbpedia will have a higher weight than links such as *rdfs:seeAlso*.

The rank scores for the different datasets are computed following the random surfer model introduced by PageRank applied to the weighted dataset graph.

To avoid the same ranking scores when performing a query towards the same dataset, DING carries out ranking of the entities within a dataset. The authors propose two different alternatives to rank entities regarding to the existence of spam within the dataset, namely EntityRank and LinkCount. For datasets with a high chance of spam, EntityRank, a weighted entity rank based on PageRank is applied. Following the same LF-IDF scheme than the dataset-ranking algorithm described above, this method guarantees robustness against spam. In case where the quality of the dataset determines the no existence of spam, LinkCount, a weighted link count algorithm can be used. Authors state that "LinkCount is more efficient to compute than EntityRank, since it needs only one iteration over the data collection". An important detail about DING is that it has been implemented with enough flexibility to consider the semantic of the dataset during the ranking of entities. DING allows the definition of a customized entity ranking algorithm that better exploits the structure of each dataset. As a justification for this approach, authors firmly affirm that "while EntityRank and LinkCount represent good generic solutions for local entity ranking, [...] an approach which takes into account the peculiar properties of each dataset will give better results". Once the dataset graph and the entity graph have been ranked, the final step is the computation of a global rank based on the combination of both scores.

5.1.12 Tonon et al.

Tonon et al. (2012) propose an algorithm for the task of Ad-hoc Object Retrieval (AOR). This task consists in retrieving entity identifiers given a query describing the entity the user is looking for, e.g., "Michael Jordan". The presented algorithm is an hybrid approach that combines content-analysis with link-analysis techniques. Specifically, the algorithm works in two steps. The initial step consists in retrieving a list of results from an inverted index using BM25F, as previously done by other authors in Pérez-Agüera et al. (2010) and Blanco et al. (2011). This index is constructed previously by storing all the information related to each entity in a document fashion. In order to keep the structure of the RDF data and improve the relevance of the ranking scores, the following pieces of information are considered:

- URI: contains the URI of the entity as found in the LOD cloud.
- Labels: this field contains datatype properties that link to textual labels with useful information about the entity. Authors state that these properties are selected previously by an independent ranking process.
- Attributes: this field considers all other datatypes properties pointing to non-label attributes.

In order to improve the recall of this step authors apply techniques like query expansion and pseudo-relevance feedback. The query expansion is implemented relying on Wordnet (Fellbaum 1998) and third parties search engines like Google. The pseudo-relevance feedback is implemented by running the initial query and then considering the labels of the top-3 retrieved entities to expand the original user query.

The limitation of the first step is that it is not able to capture the interlinked characteristic of LOD data. With this aim the second step of the algorithm exploits the graph structure of the data and help to improve the final ranking. For doing so, authors implement graph traversals using SPARQL queries. These queries are constructed using the ranking seeds from the first step. The graph traversals are executed at distances 1 and 2. Authors state that greater distances impose a higher overhead and little improvements.

The final scores are computed using a linear combination of the two steps. Authors claim an improvement of 25 % over state-of-the-art approaches.

5.2 Semantic search engines examples

Like in the traditional hypertext Web, semantic search engines are used as the entry point to navigate the information burden. The architecture of these systems is built on top of a crawler which is responsible for automatically exploring the Web of Data to retrieve any piece of semantic information (RDF, RDFa, etc.). This stream of information fed by the crawler constitutes the knowledge base that will be used to resolve the users's queries. It is in this phase where ranking algorithms play their role, by means of manipulating the information in the knowledge base to calculate the relevance of results. In the following, we describe some prototypes to show how ranking algorithms are used within semantic search engines.

Sindice Sindice (Tummarello et al. 2007) is a lookup service built with the aim of enabling information retrieval over the resources of the semantic Web. Through crawling techniques, Sindice analyzes each source of data, i.e. RDF document or SPARQL endpoint, and extracts all the resources encountered. The information related to these resources is stored in an index that can be queried based on full-text search, URIs or inverse-functional properties (IFPs). An important detail is what the authors state: "Sindice only acts as locator of RDF resources, returning pointers to remote data sources, and not as a query engine". In order to facilitate its consumption, Sindice does not compute a global ranking of all sources, but ranks the results obtained after index retrieval on query time. Sindice ranks the results according to a ranking function that takes into account the metadata associated to sources and external ranking services. The final rank value is calculated upon an unweight average of the following metrics:

1. Hostname: Sindice considers more relevant sources whose hostname is the same as the resource's hostname, in support of the linked data paradigm.
2. External rank: Sindice considers more relevant sources hosted on sites which rank high using traditional Web ranking algorithms.
3. Relevant sources: Sindice prefers sources that share rare terms (URIs, IFPs, keywords) rather than common terms with the requested terms. This relevance metric is computed through a combination of link-based analysis (DING) and BM25MF (Campinas et al. 2012).

SemSearch SemSearch (Lei et al. 2006) is a keyword-based semantic search engine that tries to bring the power of the semantic Web to all kind of users regardless of their knowledge about semantic technologies while producing accurate results at the same time. It provides a Google-like interface which ranks the search results according to the degree of their proximity to the user query. The search engine takes two factors into consideration when ranking. One is the matching distance between each keyword and its semantic matches. The other is the number of keywords the search results satisfies. The matching strategy relies on simple string comparisons between the user keywords and the labels available in the RDF data sources. Authors justify this choice stating that "from the user point of view labels often catch the meaning of semantic entities in an understandable way".

Swoogle Swoogle (Finin et al. 2004) was intended as a search engine for retrieving semantic Web documents. With this aim it is composed of a crawler than constantly checks the Web looking for new RDF or OWL documents containing any kind of semantic

information. Once the documents are found, the system uses an index to store the information and facilitate the retrieval. In the same way than traditional search engines over HTML documents, Swoogle allows users to look for any term within the indexed documents. The way how results are returned to the user is determined by the OntologyRank algorithm.

Falcons Falcons (Cheng and Qu 2009) is a keyword-based search engine supporting full-text queries related to data in the semantic Web. It works at entity level granularity, and so, for each entity it shows information about its types, possible labels and number of documents where it appears. The search engine is fully implemented relying on an index that stores textual information about each entity, as well as its relationships with other entities. Falcons applies the TF-IDF technique over the index to retrieve information about the entities and therefore about the ontologies where they appear.

Sig.ma Tummarello et al. (2010) describes the implementation of Sig.ma, an application that shows a possible interpretation of how the Web of Data functionality should look like. It combines large scale semantic web indexing, logic reasoning, data aggregation heuristics, ad hoc ontology consolidation, user interaction and refinement. Sig.ma is built on top of Sindice, which means it follows the same ranking approach. More than a search engine, it has been designed with the aim of mashing up information, i.e., it gathers information from different sources and place it in a single interface to provide a richer experience to the user. Sig.ma can be considered as an extension to Sindice, which enables a refinement towards entity-oriented search.

SWSE The Semantic Web Search Engine (Hogan et al. 2011) consists of crawling, data enhancing, indexing and a user interface for search, browsing and retrieval of information. It has been designed to deal with two main challenges: scalability to large amounts of data and tolerance to heterogeneous, noisy and conflicting data retrieved from different sources. The search performed by the SWSE is focused on entities over instance data, in contrast to other approaches like Swoogle, which follows a document oriented search over ontologies. The underlying ranking strategy of SWSE relies on ReConRank, which means that it focuses on provenance of data to establish an order for results.

Watson Watson (Sabou et al. 2007-06) is intended to be a gateway to access the content of the semantic Web. It provides keyword search facilities over semantic Web documents, but additionally provides search over entities. Authors establish that while following a traditional Web approach to retrieve information about ontologies is useful, it is not enough and must be complemented with specific techniques to exploit the semantics they model. In this way, approaches like OntologyRank that rely on the popularity of ontologies to establish the order of results are criticized, as they do not reflect the real quality of the information contained. In real scenarios, where the main aim of these systems is the reutilization, the quality of the ontology can be as important as its popularity. Therefore, Watson implements a ranking strategy similar to the one implemented by AKTiveRank, where the scores are calculated relying on exhaustive analysis to derive the quality of data.

5.3 Keyword search in databases

This section describes the most relevant approaches on keyword search. We have tried to focus the descriptions on the implementation of the scoring functions. For further details about query resolution and optimization we suggest the reader to consider the work in Coffman and Weaver (2010).

5.3.1 BANKS

BANKS (Bhalotia et al. 2002) implements keyword search on top of relational models. It uses a graph as input data model, which is created from a relational database as follows. Each tuple in the database is represented as a node. The relationships (foreign keys) are used to create the edges of the graph. For each edge a backward edge is created in order to avoid hubs in the graph. Following a similar approach to the in-link degree in PageRank (authority), nodes have weights according to the amount of references they get. The weight associated to an edge represent how close two tuples (nodes) are. By default the weight is established to 1 and the smaller this value the closer two nodes are.

In BANKS, a search consists on finding those nodes matching each of the terms that compose a keyword query. These matches are established by string comparisons of the keywords with the data and metadata (relationship and column names) available in the relational model. The answer to a query consists on a rooted directed tree containing at least a node for each term. The ranking algorithm tries to find answers which minimizes the edge weights and maximize the node weights. This problem is equivalent to the Steiner tree problem, which is known to be NP hard. Authors describe an approximation to such problem by using a *backward expanding search* algorithm. They assume that their data model can fit in memory, which may be true for moderately large databases, but cannot be considered as certain on the Web of Data.

In Kacholia et al. (2005) authors implement an extension to BANKS in which they introduce forward search. With this modification the amount of accessed nodes during the graph traversal is reduced.

5.3.2 DISCOVER

DISCOVER (Hristidis and Papakonstantinou 2002) implements a similar idea to BANKS, but using a reduced model. Authors do not include backwards links and do not assign weights to the nodes and the edges of the graph. Nevertheless, for the query resolution they implement a greedy algorithm which tries to minimize the distance between nodes. This algorithm tries to produce all possible answers for a user query without ranking the results and therefore it does not implement any kind of scoring function.

Hristidis et al. (2003) is an extension to DISCOVER in which authors implement support for boolean OR and AND keyword search. Additionally, the authors improve the performance of the algorithm by only computing the top-k results according to a scoring function based on TF-IDF. Authors in Liu et al. (2006) further develop the scoring function introducing four different options to normalize the ranking scores. They claim a 77.4 % of improvement over (Hristidis et al. 2003).

5.3.3 ObjectRank

ObjectRank (Balmin et al. 2004) extends PageRank to performs keyword search in databases. Differently from PageRank, it takes into account the semantic of relationships between the database objects assuming a certain grade of authority/importance. ObjectRank sorts the database objects with respect to a given keyword query following a flexible strategy by mean of allowing the user to adjust the system according to the domain and/or his specific requirements.

ObjectRank is applied within a system that discerns between preprocessing and query stage. In the preprocessing phase a global ranking similar to PageRank is computed. The

result of this phase is an inverted index built on the keywords available in the database schema. In the query stage, a keyword-specific ranking is calculated. The authors justify this methodology stating that "it is substantially more efficient to first calculate the global ObjectRank, and use these scores as initial values for the keyword-specific computations. This accelerates convergence, since in general, objects with high global ObjectRank, also have high keyword-specific ObjectRanks." The final score is determined as a combination of the two stages.

5.3.4 BLINKS

BLINKS (He et al. 2007) performs keyword search over graph structures. It was designed to reduce the memory consumption and low performance of methods like BANKS (Bhalotia et al. 2002). BLINKS' efficiency relies on its underlying indexing technique. It uses a two layers index for reducing the space and boosting the search. Authors rely on this index to create a variation of the *backward expanding search* called *cost-balanced expansion*, which reduces the amount of accessed nodes during the graph traversal.

BLINK implements a ranking function similar to the one described in BANKS (Bhalotia et al. 2002), which considers both content and graph structure. Authors state that their main focus is not to improve the scoring mechanism, but to improve the indexing and query processing.

6 Evaluation approaches

Performing a quantitative evaluation and comparison of the algorithms would require their complete implementation, as the source code has not been published in most of the cases. In addition, due to the different policies used to implement the ranking approaches here described, it is very difficult to establish a technical comparison to analyze the accuracy and precision of each algorithm in reference to others. Well known benchmarks in the area of information retrieval like Artiles et al. (2008), Kamps et al. (2008), Soboroff et al. (2006) are not applicable or do not cover the possible scenarios. In this section we present the existent attempts to evaluate ranking algorithms on the Web of Data.

6.1 Document oriented evaluation approaches

The lack of evaluation frameworks targeting semantic search was pointed first in workshops like the Semantic Search series.[8] Initial efforts to evaluate semantic search systems were mere adaptations of document evaluation techniques (Artiles et al. 2008; Kamps et al. 2008; Soboroff et al. 2006. This is the approach followed by authors in Fernandez et al. (2008), who rely on the TREC benchmark. The same strategy could be applied to document oriented ranking approaches like the one implemented by OntologyRank in Finin et al. (2004). However, applying this benchmarks directly have some difficulties. First, when the query cannot be handled by the ranking algorithm many authors try to compensate by implementing basic keyword search. The consequence is that it is difficult to estimate how much an algorithm is exploiting semantic information. We could argue that it would be possible to eliminate those queries that are not semantically understandable,

[8] http://km.aifb.kit.edu/ws/semsearch08/.

however different algorithms may understand different query sets. Performing an evaluation using different query sets may produce results that are not comparable.

The second problem derived from the use of document oriented benchmarking is that not all semantic ranking approaches target documents, but rather retrieve knowledge encoded in some semantic data model. In many contexts, the data in the system is not necessarily associated with any particular text document. Even in cases where there is a document, an evaluation based on document rankings is not able to measure some of the key advantages of semantic search such as being able to give precise answers to factual questions or that answers can be computed by aggregating knowledge from different documents.

6.2 Ad-hoc object retrieval evaluation approaches

Authors in Pound et al. (2010) were pioneers in defining the task of ad-hoc object retrieval and proposing a methodology for its evaluation. The task consists on answering certain information needs related to particular aspects of objects, expressed using plain natural language (i.e., avoiding any kind of formal query language) and resolved using a collection of structured data. Within this task authors established five different query categories that determine the behavioral needs of the underlying ranking approach:

- *Entity query* the expected result is a particular entity or list of entities.
- *Type query* expected results are entities that are instances of certain type or the URI identifier of the type itself.
- *Attribute query* expected results are values of a particular attribute associated to an entity or type.
- *Relation query* expected results are relationships among entities or types.
- *Other keyword query* this category includes those queries that do not fit any of the previous categories.

From the above categories, entity retrieval has got special attention as can be observed in the latest ranking contributions discussed in this work. Existing efforts on the evaluation of this kind of ranking methodologies seem to consolidate and start focusing on the ranking of entities. The INEX 2007–2009 Entity Retrieval track (Vries et al. 2008; Demartini et al. 2009, 2010 studies entity retrieval using an XML corpus bases on Wikipedia. In the INEX 2009 this corpus was enriched with semantic knowledge from YAGO (Suchanek et al. 2007; chenkel and Kasneci 2007). The Linked Data track at INEX 2012 (Wang et al. 2012) also considers entities from Wikipedia, but articles are enriched with RDF properties from both DBpedia and YAGO2 (Hoffart et al. 2011). The INEX track also includes textual corpora and its main target is to measure how structured data can improve the retrieval performance. This means that systems using this corpus compete also on information extraction functionality. The TREC 2009–2011 Entity track (Balog et al. 2009, 2010, 2011) defines the related entity finding task (REF): return homepages of entities, of a specified type, that engage in a specified relationship with a given source entity. As an extension to the REF task, the entity list completion task (ECL) was introduced in TREC 2010. The main difference to REF is that entities are not represented by their home pages, but by a unique URI from a specific collection (in particular the Billion Triple Challenge 2009 collection[9]). In 2010, the Semantic Search Challenge[10] (Halpin et al. 2010)

[9] http://vmlion25.deri.ie/.

[10] http://semsearch.yahoo.com/.

introduced a platform for evaluating ad-hoc queries, targeting a particular entity. The queries were constructed from a Yahoo search query log and the data corpus was again the Billion Triple Challenge 2009 dataset. The 2011 edition of the challenge presented a second task based on list search. The goal of this track was to select a list of entities matching particular criteria. In 2012 there was not Semantic Search Challenge edition, but a joint workshop on entity search (Balog et al. 2012). As a result of this workshop, authors in Balog and Neumayer (2013) propose a new entity search test collection based on DBpedia data. The test collection includes a mix of queries from the previous benchmarking evaluation campaigns together with corresponding relevance judgements. The authors provide baseline results based on language models and BM25. This test set can be considered as the first effort to consolidate the existent benchmarks for evaluating entity oriented ranking approaches. All resources related to this benchmark can be found at http://krisztianbalog.com/resources/sigir-2013-dbpedia/.

Finally, at the time of writing, we could not identify any community approach to evaluate ranking algorithms that target relationships among entities, i.e., Anyanwu et al. (2005) and Franz et al. (2009). We do not foresee further research in this direction due to the traction of entity oriented ranking approaches.

6.3 Keyword search evaluation approaches

In parallel to entity retrieval, the database community has identified the need for standardizing the evaluation of keyword search approaches. As stated in Coffman and Weaver (2010): "the strategic step of creating a DB&IR evaluation forum has yet to occur. Without it, progress will not match that of the larger IR community." In the same work, authors describe a benchmark for comparing the performance of keyword search approaches for databases. The benchmark includes data derived from three different datasets, namely DBLP, IMDb and MONDIAL (May 1999). Additionally, it includes a set of 150 queries identifying different information needs. Every query includes a relevance assessment for computing the ranking metrics. After applying this benchmark to nine of the state-of-the-art approaches, authors conclude that current approaches do not scale enough to cover size-demanding databases and call the community to produce algorithms to support a bigger workload. The benchmark data is publicly available at http://www.cs.virginia.edu/~jmc7tp/projects/search/.

7 Conclusions

In this work, we have presented the results of the research carried out on the field of ranking strategies in the last years. In particular, we have focused in the case of the Web of Data. Looking into details, it can be appreciated a certain grade of parallelism between the development of these techniques and the evolution of the Semantic Web. While the first ranking algorithms were designed with the aim of ranking ontology documents, in a similar way to how traditional approaches rank HTML documents, the arrival of the Web of Data has changed the focus of ranking strategies towards information modeled as entities and their relationships. Additionally, we have reviewed the parallel efforts going on in the database community through keyword search approaches. Based on this review, we have discussed open challenges and establish the base for new research directions.

We have formalized the problem of ranking information and shown its relevancy for data management and consumption. We have also discussed a possible classification and

unified the core concepts that characterize ranking algorithms. We have used this classification to discuss and analyze a variety of ranking implementations that summarize the research trajectory of the last 10 years.

We have provided an overview of the current approaches pursuing the standardization of the efforts to evaluate ranking algorithms.

The analysis reported in this survey aims at providing a starting point towards future developments on benchmarking and empirical evaluation of ranking solutions for the Web of Linked Data. We also hope that this survey will be useful to programmers who need to implement a ranking algorithm and to researchers considering the design of new ranking strategies.

References

Dbpedia spotlight. (2011). Shedding light on the web of documents. In *In the proceedings of the 7th international conference on semantic systems (I-Semantics)* .

Alani, H., Brewster, C., & Shadbolt, N. (2006). Ranking ontologies with aktiverank. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Eds.), *International semantic web conference, lecture notes in computer science* (Vol. 4273, pp. 1–15). Berlin: Springer.

Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Void guide—using the vocabulary of interlinked datasets. http://rdfs.org/ns/void-guide.

Anyanwu, K., Maduko, A., & Sheth, A. P. (2005). Semrank: Ranking complex relationship search results on the semantic web. In A. Ellis & T. Hagino (Eds.), WWW, pp. 117–127. ACM.

Artiles, J., Sekine, S., & Gonzalo, J. (2008). Web people search: Results of the first evaluation and the plan for the second. In WWW, pp. 1071–1072.

Baeza-Yates, R., & Davis, E. (2004). Web page ranking using link attributes. In: Proceedings of WWW-04and the 13th international World Wide Web conference—alternate track papers & posters, pp. 328–329. ACM Press.

Balmin, A., Hristidis, V., & Papakonstantinou, Y. (2004). Objectrank: Authority-based keyword search in databases. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, K. B. Schiefer (Eds.), VLDB, pp. 564–575. Morgan Kaufmann.

Balog, K., Carmel, D., de Vries, A. P., Herzig, D. M., Mika, P., Roitman, H., et al. (2012). The first joint international workshop on entity-oriented and semantic search (jiwes). *SIGIR Forum*, *46*(2), 87–94.

Balog, K., & Neumayer, R. (2013). A test collection for entity search in dbpedia. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, SIGIR '13*, pp. 737–740. ACM, New York, NY. doi:10.1145/2484028.2484165.

Balog, K., Serdyukov, P., & de Vries, A. P. (2010). Overview of the trec 2010 entity track. In *TREC*.

Balog, K., Serdyukov, P., & de Vries, A. P. (2011). Overview of the trec 2011 entity track. In *TREC*.

Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., & Westerveld, T. (2009). Overview of the trec 2009 entity track. In *TREC*.

Berners-Lee, T. (2006). Linked data—design issues. http://www.w3.org/DesignIssues/LinkedData.html.

Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., & Sudarshan, S. (2002). Keyword searching and browsing in databases using banks. In *ICDE*, pp. 431–440. IEEE Computer Society. http://dblp.uni-trier.de/rec/bibtex/conf/icde/BhalotiaHNCS02.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data—the story so far. *International Journal on Semantic Web and Information Systems*, *5*(3), 1–22.

Blanco, R., Mika, P., & Vigna, S. (2011). Effective and efficient entity search in rdf data. In *Proceedings of the 10th international conference on The semantic web—volume part I, ISWC'11* (pp. 83–97). Berlin, Heidelberg: Springer. http://dl.acm.org/citation.cfm?id=2063016.2063023.

Brickley, D., & Guha, R. (2014). Rdf vocabulary description language 1.1: Rdf schema—w3c recommendation. http://www.w3.org/TR/rdf-schema/.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, *30*(1–7), 107–117.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the web. *Computer Networks*, *33*(1–6), 309–320.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Campinas, S., Delbru, R., & Tummarello, G. (2012). Effective retrieval model for entity with multi-valued attributes: Bm25mf and beyond. In *EKAW*, pp. 200–215.

Chen, N., & Prasanna, V. K. (2012). Learning to rank complex semantic relationships. *IJSWIS*, *8*(4), 1–19.

Cheng, G., & Qu, Y. (2009). Searching linked objects with falcons: approach, implementation and evaluation. *International Journal on Semantic Web and Information System*, *5*(3), 49–70.

Coffman, J., & Weaver, A. C. (2010). A framework for evaluating database keyword search strategies. In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, A. An (Eds.), *CIKM*, pp. 729–738. ACM. http://dblp.uni-trier.de/db/conf/cikm/cikm2010.html#CoffmanW10

Cyganiak, R., Harth, A., & Hogan, A. (2008). N-quads: Extending n-triples with context. http://sw.deri.org/2008/07/n-quads/.

Dali, L., Fortuna, B., Tran, D. T., & Mladenic, D. (2012). Query-independent learning to rank for rdf entity search. In *ESWC*, pp. 484–498.

Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., & Decker, S. (2010). Hierarchical link analysis for ranking web data. In *Proceedings of the 7th international conference on the semantic web: Research and applications—volume part II, ESWC'10* (pp. 225–239). Berlin, Heidelberg: Springer.

Demartini, G., Iofciu, T., De Vries, A. P. (2010). Overview of the inex 2009 entity ranking track. In *Proceedings of the focused retrieval and evaluation, and 8th international conference on Initiative for the evaluation of XML retrieval, INEX'09* (pp. 254–264). Berlin, Heidelberg: Springer. http://dl.acm.org/citation.cfm?id=1881065.1881096.

Demartini, G., Vries, A. P., Iofciu, T., & Zhu, J. (2009). *Advances in focused retrieval. chap. Overview of the INEX 2008 entity ranking track* (pp. 243–252). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-03761-0_25.

Fellbaum, C. (1998). A semantic network of english: the mother of all wordnets. *Computers and the Humanities*, *32*(2–3), 209–220.

Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., et al. (2008). Semantic search meets the web. In *Proceedings of the 2008 IEEE international conference on semantic computing, ICSC '08* (pp. 253–260). IEEE Computer Society, Washington, DC, USA. doi:10.1109/ICSC.2008.52.

Finin, T., Peng, Y., Scott, R., Joel, C., Joshi, S. A., Reddivari, P., et al. (2004). Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM conference on information and knowledge management* (pp. 652–659). ACM Press.

Franz, T., Schultz, A., Sizov, S., & Staab, S. (2009). Triplerank: Ranking semantic web data by tensor decomposition. In *International semantic web conference (ISWC)*.

Franz, T., Schultz, A., Sizov, S., & Staab, S.(2009). Triplerank: Ranking semantic web data by tensor decomposition. In *International semantic web conference* (pp. 213–228).

Getoor, L., & Diehl, C. P. (2005). Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, *7*(2), 3–12.

Halpin, H., Herzig, D. M., Mika, P., Blanco, R., Pound, J., Thompson, H. S., et al. (2010). Evaluating ad-hoc object retrieval. In *Proceedings of the international workshop on evaluation of semantic technologies (IWEST 2010). 9th international semantic web conference (ISWC2010)*, Shanghai, PR China.

Harth, A., Kinsella, S., & Decker, S. (2009). Using naming authority to rank data and ontologies for web search. In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, et al. (Eds.), *International semantic web conference, lecture notes in computer science* (Vol. 5823, pp. 277–292). Berlin: Springer.

He, H., Wang, H., Yang, J., & Yu, P. S. (2007). Blinks: Ranked keyword searches on graphs. In *SIGMOD '07. Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 305–316). New York, NY: ACM Press. doi:10.1145/1247480.1247516.

Hildebrand, M., van Ossenbruggen, J., & Hardman, L. (2007). *An analysis of search-based user interaction on the semantic web*. Nederlands, Centrum voor Wiskunde en Informatica: Tech. rep.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., de Melo, G., Weikum, G. (2011). Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web, WWW '11* (pp. 229–232). New York, NY: ACM. doi:10.1145/1963192.1963296.

Hogan, A., Harth, A., & Decker, S. (2006). Reconrank: A scalable ranking method for semantic web data with context. In *In 2nd workshop on scalable semantic web knowledge base systems*.

Hogan, A., Harth, A., Umrich, J., Kinsella, S., Polleres, A., & Decker, S. (2011). Searching and browsing linked data with swse: The semantic web search engine. *Journal of Web Semantics*, *9*(4), 365–401.

Hristidis, V., Gravano, L., & Papakonstantinou, Y. (2003). Efficient ir-style keyword search over relational databases. In *VLDB*, pp. 850–861. http://dblp.uni-trier.de/db/conf/vldb/vldb2003.html#HristidisGP03.

Hristidis, V., & Papakonstantinou, Y. (2002). Discover: Keyword search in relational databases. In *VLDB*, pp. 670–681. Morgan Kaufmann.

Jansen, B., & Spink, A. (2006). How are we searching the world wide web? A comparison of nine search engine transaction logs. *Information Processing and Management*, *42*(1), 248–263.

Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., & Karambelkar, H. (2005). Bidirectional expansion for keyword search on graph databases. In K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P. Å. Larson & B.C. Ooi (Eds.), *VLDB*, pp. 505–516. ACM. http://dblp.uni-trier.de/db/conf/vldb/vldb2005.html#KacholiaPCSDK05.

Kamps, J., Geva, S., Trotman, A., Woodley, A., & Koolen, M. (2008). Overview of the inex 2008 ad hoc track. In *INEX*, pp. 1–28.

Kasneci, G., Suchanek, F. M., Ifrim, G., Ramanath, M., Weikum, G. (2008). Naga: Searching and ranking knowledge. In: G. Alonso, J. A. Blakeley & A. L. P. Chen (Eds.), *ICDE*, pp. 953–962. IEEE. http://dblp.uni-trier.de/db/conf/icde/icde2008.html#KasneciSIRW08.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th annual ACM-SIAM symposium on discrete algorithms*.

Klyne, G., & Carroll, J. (2004). Resource description framework (rdf): Concepts and abstract syntax—w3c recommendation. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210.

Lassila, O. (2007). Programming semantic web applications: A synthesis of knowledge representation and semi-structured data. Ph.D. thesis, Helsinki University of Technology.

Lei, Y., Uren, V. S., & Motta, E. (2006). A search engine for the semantic web. In *EKAW, lecture notes in computer science* (pp. 238–245). Semsearch: Springer.

Lempel, R., & Moran, S. (2001). Salsa: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems*, *19*(2), 131–160.

Liu, F., Yu, C. T., Meng, W., Chowdhury, A. (2006). Effective keyword search in relational databases. In S. Chaudhuri, V. Hristidis & N. Polyzotis (Eds.), *SIGMOD conference*, pp. 563–574. ACM. http://dblp.uni-trier.de/db/conf/sigmod/sigmod2006.html#LiuYMC06.

Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–331. doi:10.1561/1500000016.

Liu, X., Croft, W. B. (2005). Statistical language modeling for information retrieval. *ARIST, 39*(1), 1–31. http://dblp.uni-trier.de/db/journals/arist/arist39.html#LiuC05.

May, W. (1999). Information extraction and integration with florid: The mondial case study. Tech. Rep. 131, Universitaet Freiburg, Institut fuer Informatik.

McGuinness, D., & van Harmelen, F. (2004). Owl web ontology language—w3c recommendation. http://www.w3.org/TR/owl-features/.

Mirizzi, R., Ragone, A., Noia, T. D., & Sciascio, E. D. (2010). Ranking the linked data: The case of dbpedia. In B. Benatallah, F. Casati, G. Kappel, & G. Rossi (Eds.), *ICWE, lecture notes in computer science* (pp. 337–354). Berlin: Springer.

Nie, Z., Zhang, Y., Wen, J. R., Ma, W. Y. (2005). Object-level ranking: Bringing order to web objects. In A. Ellis & T. Hagino (Eds.), WWW, pp. 567–574. ACM.

Pérez-Agüera, J. R., Arroyo, J., Greenberg, J., Iglesias, J. P., & Fresno, V. (2010). Using bm25f for semantic search. In *Proceedings of the 3rd international semantic search workshop, SEMSEARCH '10* (pp. 2:1–2:8). New York, NY: ACM. doi:10.1145/1863879.1863881.

Pound, J., Mika, P., & Zaragoza, H. (2010). Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web, WWW '10* (pp. 771–780). New York, NY: ACM.

Roa-Valverde, A. J. (2011). Multimedia information retrieval as a practical application for interlinking approaches. In *Proceedings of the 7th international conference on semantic systems, I-Semantics '11* (pp. 230–233). New York, NY, USA: ACM.

Sabou, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Motta, E., d'Aquin, M., et al. (2007–06). Watson: A gateway for the semantic web. In *ESWC 2007 poster session*.

Sawant, U., & Chakrabarti, S. (2013). Features and aggregators for web-scale entity search. CoRR abs/1303.3164.

Schenkel, F. S. R., & Kasneci, G. (2007). Yawn: A semantically annotated wikipedia xml corpus. http://www.mpi-inf.mpg.de/%7Ekasneci/download/BTW2007.pdf.

Sheth, A., Arpinar, I., & Kashyap, V. (2004). Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships. In M. Nikravesh, B. Azvine, R. Yager & L. Zadeh (Eds.), Enhancing the power of the internet, studies in fuzziness and soft computing, vol. 139, pp. 63–94. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-45218-8_4.

Sicilia, M. Á., Rodríguez, D., Barriocanal, E. G., & Alonso, S. S. (2012). Empirical findings on ontology metrics. *Expert Systems with Application*, *39*(8), 6706–6711.

Soboroff, I., de Vries, A.P., & Craswell, N. (2006). Overview of the trec 2006 enterprise track. In *TREC*.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pp. 697–706. New York, NY, USA: ACM. doi:10.1145/1242572.1242667.

Tonon, A., Demartini, G., & Cudré-Mauroux, P. (2012). Combining inverted indices and structured search for ad-hoc object retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '12*, pp. 125–134. New York, NY, USA: ACM. doi:10.1145/2348283.2348304.

Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., & Decker, S. (2010). Sig.ma: Live views on the web of data. *Journal of Web Semantics*, *8*(4), 355–364.

Tummarello, G., Oren, E., & Delbru, R. (2007). Sindice.com: Weaving the open linked data. In *Proceedings of the 6th international semantic web conference and 2nd Asian semantic web conference (ISWC/ASWC2007)* (vol. 4825, pp. 547–560). Busan, South Korea, LNCS. Berlin, Heidelberg: Springer.

Vries, A. P., Vercoustre, A. M., Thom, J. A., Craswell, N., & Lalmas, M. (2008). *Focused access to xml documents. Chap. Overview of the INEX 2007 entity ranking track*, pp. 245–251. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-85902-4_22.

Wang, Q., Kamps, J., Ramirez Camps, G., Marx, M., Schuth, A., Theobald, M., et al. (2012). Overview of the INEX 2012 linked data track. In P. Forner, J. Karlgren & C. Womser-Hacker (Eds.), CLEF 2012 evaluation labs and workshop: Online working notes, pp. 1–13. Rome, Italy.

Wei, W. (2009). Semantic search: Bringing semantic web technologies to information retrieval. Ph.D. thesis, University of Nottingham.

Xing, W., & Ghorbani, A. A. (2004). Weighted pagerank algorithm. In *CNSR*, pp. 305–314. IEEE Computer Society.

Xue, G. R., Yang, Q., Zeng, H. J., Yu, Y., & Chen, Z. (2005). Exploiting the hierarchical structure for link analysis. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 186–193). New York, NY, USA: ACM Press.

Yu, J. X., Qin, L., & Chang, L. (2010). Keyword search in relational databases: A survey. *IEEE Data Engineering Bulletin*, *33*(1), 67–78.

Zhu, X., Goldberg, A. B., Van, J., & Andrzejewski, G. D.(2007). Improving diversity in ranking using absorbing random walks. In *Physics laboratory—University of Washington*, pp. 97–104.