Noname manuscript No.
(will be inserted by the editor)

A Survey of Approaches for Ranking on Structured Data

First Author · Second Author

Received: date / Accepted: date

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 follow. 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. Elaborating a technical comparison of ranking strategies is a difficult task due to the heterogeneity of approaches and the diversity of data used for the task. Further, in many cases, the source code of the implementations is not available, but just a methodology describing the behavior of a proposed algorithm. However, a thorough analysis is still required to understand the advantages and drawbacks of the different approaches. In order to fulfill this need, this survey proposes a reference framework 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.

Keywords Linked data · information retrieval · semantic search · ranking algorithms · link analysis · Semantic Web data management

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

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 & 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 & 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 in (Sheth, Bu-

dak Arpinar & Kashyap, 2004). 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, Heath, & Berners-Lee, 2009) as a Web of things in the world, in contrast to the traditional abovementioned Web of documents. 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 machines 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 & Carroll, 2004), RDF Schema (McGuinness & van Harmelen, 2004) and the Web Ontology Language (Brickley & Guha, 2004) 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, Heath, & Berners-Lee, 2009) “Linked Data is simply about using the Web to create typed links between data from different sources”. 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 (i) to assign consistent URIs to data published, (ii) to generate links, and (iii) 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, Heath, & Berners-Lee, 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 & Page, 1998), HITS (Kleinberg, 1998) and SALSA (Lempel & Moran, 2000), 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, van Ossenbruggen & Hardman 2007; Mkel, 2007; Want, 2008), 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

[1] http://linkeddata.org/

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.

Outline

2 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 $a, b \in X : f(a) \leq f(b) \leftrightarrow a \prec b$, where $\prec$ defines a binary relationship on the set X. Note that $\prec$ makes reference to the factor that guides the ranking strategy.

Figure 1 depicts the components that integrate the functionality of a ranking algorithm. The input to any ranking method is a bunch of raw data that needs to be inspected to get the relevant information out of it. This task is carried out by the feature extractor, which generates a data model that will be used for the ranking function described above to produce the ranked set of items. The feature extractor together with the data structures that allocate the data model and the ranking function compose the functional architecture of the ranking algorithm.

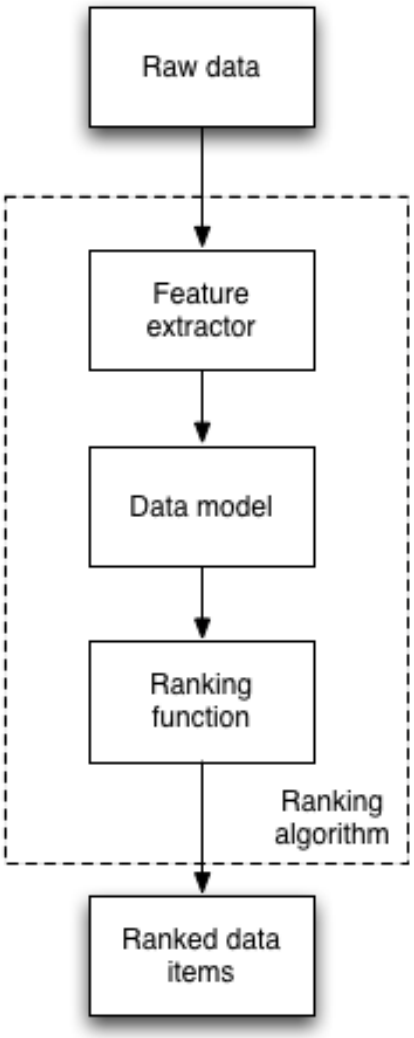

Fig. 1: **Structure of a ranking algorithm**

2.1 Ranking on Structured Data

Structured data: can be conceived as graphs
Ranking: traditionally, core IR problem, many techniques have been adopted to deal with structured data
Ranking over structured data: ranked list of results
Types of ranked results: entities, relations, subgraphs, entire datasets (graphs)
Evaluating ranked results: different metrics

2.2 Ranking on Structured Data vs. Ranking on Unstructured Data

Discuss differences
Say we focus on structured data

2.3 Ranking vs. Matching

Discuss differences
Say we focus on ranking

2.4 Applications

Using information is directly related to the way in which it is presented and consumed. This means that without methodologies to exploit properly the data it can be very difficult to deal with the huge amount of information available. This function is carried out by ranking algorithms and a proof of its applicability is the different use cases where they can be deployed. In the following, we describe how ranking al-

gorithms are used within semantic search engines, semantic browsers and interlinking approaches.

2.4.1 Semantic Search Engines

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. Table **??** shows the existent semantic search engines and the correspondence with the underlying implemented ranking algorithm.

TODO table with sse

In the following, we describe the approach followed by each search engine.

Sindice Sindice (Tummarello, Delbru, Oren, 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).

SemSearch SemSearch (Lei, Uren, Motta, 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 (Ding 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, & 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 (Tumarello et. al, 2009) 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, 2010) unlike other search engines works over Linked Open Data. It 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 (dAquin et. al, 2007) 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.

2.4.2 Semantic Browsers

Browsers are the tool utilized by users to access the information on the Web. This means that the way how the information is presented to the users is a decisive factor for its consumption, because it determines the next steps for the user while the exploration. In the same way than traditional browsers explore the links between hypertext documents, semantic browsers intend to explore links between RDF data. Tabulator (Berners-Lee et al. 2006; Berners-Lee et al. 2008) and Marble (Becker & Bizer, 2008) are two examples of semantic browsers that apply ranking methodologies to analyze the provenance of data at the same time that merging information from different data sources.

2.4.3 Interlinking

Interlinking can be defined as the process of creating new links between related entities in different RDF datasets for which there are not any link previously established. The final aim of interlinking is creating a network of related entities in order to facilitate its discovery and navigation of the Web of Data. As introduced in (Bizer et. al, 2009), automatic and semi-automatic interlinking approaches rely on the computation of similarities between entities, mostly guided by related work on record linkage, duplicate detection in databases and ontology matching. However, what is not directly mentioned is that ranking algorithms can complement interlinking methods by providing previous analysis of data. Statistics about entities can give information about the truthfulness of concepts as stated in (Bizer & Cyganiak, 2009). For example, if a concept representing the city of Barcelona is pointed from different sources where properties make reference to this city, it means there is a chance that this concept really represents the city of Barcelona. In this way, the number of links targeting the same concept could be used as an indicator of the veracity of such concept when referring to an entity of the real world.

3 Ranking Approaches

3.1 Historical Development

3.2 Generic Architecture

3.3 Main Dimensions

Data: structured data conceived as graphs? All existing approaches are applicable to data graphs or are there any that exploit special characteristics of specific types of structured data? Only in the latter case, approaches shall be distinguished along this data dimension

Queries: same as data dimension: are there many different approaches dealing with different types of queries so that we can use query as a dimension to distinguish approaches?

Features: content, structure, context, two meanings of context: context of user; and context of the data such as trust values of source, truth value / confidence degree of individual records; make clear this survey does not discuss user-context based methods in details

Results: entity, relationships, subgraphs, graphs (entire dataset)

Techniques:
- may vary in terms of methods: used *NLP* for understanding keywords, used *ML* for classifying keywords (I know some ML-based query classification approaches for document retrieval, are there also examples for structured data ranking?) and for learning to rank, (other) statistics-based methods: Vector Space Model (VSM), Language Modeling (LM), Information Theory
- may vary in terms of heuristics: two main ones are query-relevance and popularity; other heuristics are proximity, informativeness (based on Information Theory, e.g. entropy) and context-based: trust, truth value, locality etc.

3.4 Foundation

Here we discuss all basics needed to understand the approaches presented in the following sections.
Vector-Space Model: also discuss pivoted normalization and point out problems of short document length etc. in the context of structured data
Language Modeling: also discuss smoothing strategies
Link Analysis
Learning to Rank
...

3.5 Taxonomy of Ranking Approaches

Classify approaches mainly based on the type of *heuristics* they used, i.e. (1) Query-relevance, (2) Popularity, (3) Other Heuristics and (4) All Heuristics / LTR.

Those that use same heuristics are distinguished in terms of *methods*. E.g. query-relevance based solutions can be further distinguished in terms of VSM-based and LM-based approaches.

an

4 Query-relevance

5 Popularity

Wikipedia defines popularity[2] as the quality of being well-liked or common, or having a high social status. Translating this to information systems, popularity ranking algorithms rely on link analysis to compute the relevance of a node of the graph in terms of its input and output linkage. 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 (statistics-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 & Page, 1998), HITS [](Kleinberg, 1998) and SALSA [](Lempel & Moran, 2000), 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 to deal with structured and semi-structured information[3], 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 & Ghorbani, 2004) and [](Baeza-Yates & 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 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 [](Ding et al., 2004) and [](Anyanwu, Maduko, & Sheth, 2005), being the last one the culmination of the ideas previously introduced in [](Anyanwu & Sheth, 2002) and [](Anyanwu & Sheth, 2003). Examples of early commercial applications exploiting semantic Web link analysis include the works described in [](Sheth, Avant & Bertram, 2001), [](Avant et. al, 2002) and [](Sheth, 2005).

6 Other Heuristics

7 Combining heuristics

Hybrid heuristics vs composite heuristics

Within composite heuristics can find manual composition vs automatic composition (i.e. Learning to rank). This taxonomy is similar to the example pointed by Tran about matching approaches.

8 Sample Approaches from the Literature

Before, we discuss the concepts/techniques behind existing approaches. Here we provide a sample

TODO: adapt table to new taxonomy

9 Open Research Challenges

10 Conclusion

For consistency, we use bibtex entries from dblp, e.g. `http://www.dblp.org/rec/bibtex/conf/sigmod/BalminKT12`. Many bibtex entries are already in the paper.tex file so just searched for these entries there first and add only new entries when needed to avoid redundancy.

References

[2] http://en.wikipedia.org/wiki/Popularity

[3] This classification has been taken from [](Delbru, Toupikov, Catasta, Tummarello & Decker, 2010)

Algorithm	**Reference**
OntologyRank	Ding et al., 2004 [**?**]
ObjectRank	Balmin, Hristidis, & Papakonstantinou, 2004 [**?**]
PopRank	Nie, Zhang, Wen & Ma, 2005 [**?**]
SemRank	Anyanwu, Maduko & Sheth, 2005 [**?**]
ReConRank	Hogan, Harth & Decker, 2006 [**?**]
AKTiveRank	Alani, Brewster & Shadbolt, 2006 [**?**]
Naming Authority Rank	Harth, Kinsella & Decker, 2009 [**?**]
TripleRank	Franz, Schultz, Sizov & Staab, 2009 [**?**]
RareRank	Wei, 2009 [**?**]
DBpediaRanker	Mirizzi, Ragone, Di Noia & Di Sciascio, 2010 [**?**]
DING	Delbru, Toupikov, Catasta, Tummarello & Decker, 2010 [**?**]

Table 1: Summary of semantic ranking algorithms and references

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

Table 2: Summary of ranking algorithms and data features