

Using Naming Authority to Rank Data and Ontologies for Web Search*

Andreas Harth**, Sheila Kinsella, and Stefan Decker

National University of Ireland, Galway
Digital Enterprise Research Institute

Abstract. The focus of web search is moving away from returning relevant documents towards returning structured data as results to user queries. A vital part in the architecture of search engines are link-based ranking algorithms, which however are targeted towards hypertext documents. Existing ranking algorithms for structured data, on the other hand, require manual input of a domain expert and are thus not applicable in cases where data integrated from a large number of sources exhibits enormous variance in vocabularies used. In such environments, the authority of data sources is an important signal that the ranking algorithm has to take into account. This paper presents algorithms for prioritising data returned by queries over web datasets expressed in RDF. We introduce the notion of naming authority which provides a correspondence between identifiers and the sources which can speak authoritatively for these identifiers. Our algorithm uses the original PageRank method to assign authority values to data sources based on a naming authority graph, and then propagates the authority values to identifiers referenced in the sources. We conduct performance and quality evaluations of the method on a large web dataset. Our method is schema-independent, requires no manual input, and has applications in search, query processing, reasoning, and user interfaces over integrated datasets.

1 Introduction

More and more structured interlinked data is appearing on the web, in the form of microformats, XML, and RDF (Resource Description Format). A common feature of these formats is that they take a loosely object-oriented view, describing objects such as people, events, or locations. Given that the information published in these formats exhibits more structure and a fine-grained description of objects, typical keyword-based search engines do not exploit the full potential that the more structured data offers. The established methods for information retrieval are not directly applicable to structured data, since i) the basic result units of search moves from documents to objects which may be associated with several sources and ii) the users are able, in addition to keyword searches, to more accurately state their information needs via precise queries.

* This work is partly funded by Science Foundation Ireland (SFI) project Lion-2 (SFI/08/CE/I1380).

** Current affiliation: Institut AIFB, Universität Karlsruhe (TH), Germany

The problem of search in hypertext collections has been extensively studied, with the web as the premier example. Given the large number of data providers compared to traditional database scenarios, an information retrieval system for hypertext on the web must be able to handle data with the following properties:

- Domain variety: the web contains data about many topics (e.g., from social networks to protein pathways to entertainment)
- Structural variety: aggregating data from many autonomous web sources, with no data curation or quality control of any sort, results in datasets with overlapping, redundant, and possibly contradictory information
- Noise: with the number of data contributors the amount of errors increase (e.g., syntax errors, spelling mistakes, wrong identifiers)
- Spam: when everybody can say anything about anything with little effort or cost, malicious activity emerges
- Scale: identifiers and documents on the web number in the trillions

We expect the properties identified above to also hold for structured information sources on the web, though they are typically not taken into account for classical relational query processing or data warehousing, where the number of autonomous sources is orders of magnitude lower. In traditional data integration systems, the schema of the integrated data is known in advance, and is hard-coded into applications to, for example, determine the order in which data elements are displayed. In this paper we focus on the problem of ranking in structured datasets which have been integrated from a multitude of disparate sources without a-priori knowledge on the vocabulary used. We assume a basic interlinked data model, enabling the definition of objects and relationships between those objects. Further, we assume the possibility of global identifiers which enable the reuse of identifiers across sources and thus the interlinkage of data. RDF, and to a limited extent XML and microformats fulfil these assumptions.

Based on the above scenario the contributions of this paper are as follows:

- We introduce the notion of naming authority which establishes a connection between an identifier (in our case a URI) and the source which has authority to assign that identifier (in our case a web source, also identified via a URI). The notion of naming authority can be generalised to other identifier schemes (e.g., trademarks) for which it is possible to establish a connection to the provenance of the identifier, such as a person or an organisation (Section 4).
- We present a method to derive a naming authority matrix from a dataset, and use the PageRank algorithm to determine rankings for sources. In a second step, an algorithm ranks individual identifiers based on the values assigned to their sources (Section 5).
- We provide an experimental evaluation on real-world web datasets containing up to 1.1bn data items from 6.5m web sources, and provide evidence for the quality of the rankings with a user study of 36 participants (Section 6).

We present an example scenario in Section 2, and provide an overview in Section 3. Section 7 outlines related approaches and Section 8 concludes.

2 Motivating Example

For the motivating example, we use social network information describing people, who they know and what they do, as there is large amounts of this data available. Data sources typically express person-related information in the Friend-of-a-Friend (FOAF) vocabulary, but also use their own schemas (i.e. sub-classing the general Person class with classes such as Professor or Ph.D. student). The personal data available on the web exhibits the properties we expect from any large-scale loosely coordinated distributed knowledge base.

In our running example URIs are used to identify both the objects (e.g. `http://danbri.org/foaf.rdf#danbri`) and the data sources (e.g. `http://danbri.org/foaf.rdf`). Identifiers might be reused across sources, for example, the source `http://danbri.org/foaf.rdf` uses the URI `http://www.w3.org/People/Berners-Lee/card#i` to denote the Person Tim Berners-Lee. The reuse of identifiers provides a direct means to consolidate information about objects from different sources. In our experimental dataset, Dan's URI is referenced in at least 70 sources. Representing only a small subset of the available data, the graphs in Figure 1 depict object descriptions from two sources.

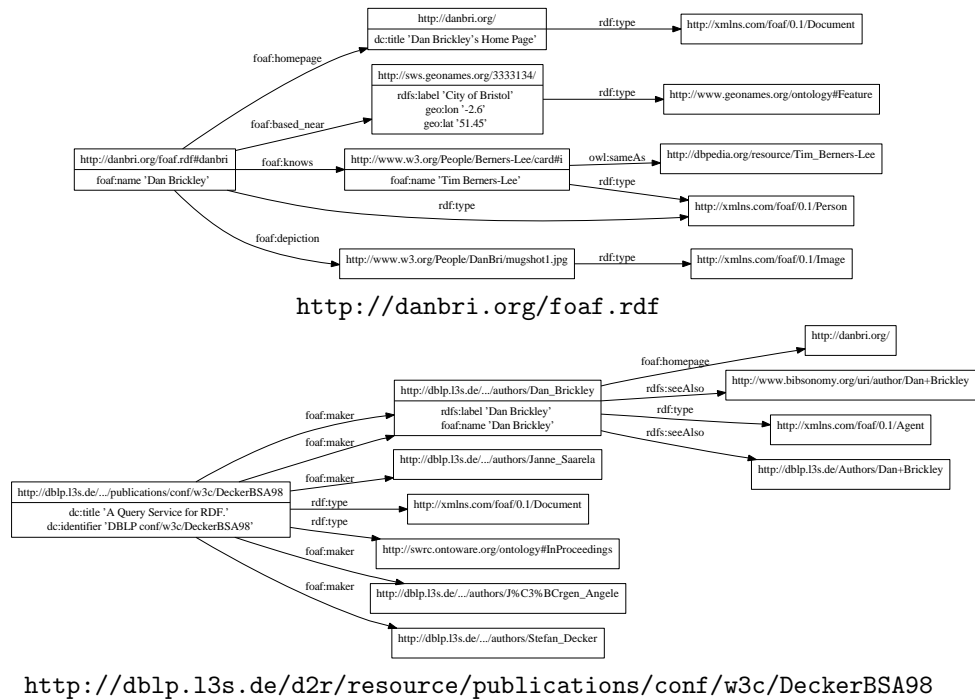


Fig. 1. A subset of RDF descriptions about Dan, taken from two sources.

Several challenges arise when displaying data aggregated from the web:

1. How to prioritise results from keyword searches (i.e. in which order to show the resulting objects)?
2. How to prioritise predicates (i.e. which predicate/value pairs to show first)?
3. How to prioritise objects from multi-valued attributes (i.e. which image to show first in case there are multiple depictions of the person)?
4. How to prioritise sources that contribute data to an object (i.e. in which order to present the sources, from the most important to the least)?

Question 1) surfaces mainly in a web search scenario, where a search engine returns a list of identifiers matching the user-specified query (keywords or structured query such as “return all instance of type `owl:Class`”) “or “return all instances of type `foaf:Person` that have `foaf:workplaceHomepage http://www.deri.org/`”. Questions 2) to 4) are also relevant for Linked Data browsers such as Disco¹, Data Explorer², or Tabulator³ in case vocabularies which are unknown to the browsers are used in the data. Figure 2 shows the rendering of information about an object using rankings derived with the method presented here in VisiNav⁴, which also has Linked Data browsing functionality.

3 Method Overview

We introduce the data model, present the requirements for a ranking method on web data and outline our procedure for ranking data sources and data items.

3.1 Data Model

In general, our ranking method can be applied to datasets with i) global, reused identifiers, ii) tracking of provenance, and iii) correspondence between object identifiers and source identifiers. Specifically, we assume the following:

- a set of identifiers I , encompassing a set of global identifiers U , a set of local identifiers B , and a set of strings L (we omit datatypes such as integer or date for brevity)
- a set of data sources $S \subseteq U$
- a function ids which maps sets of global and local identifiers and literals $i \in I$ to the sources $s \in S$ in which they occur

This generic model applies to a wide variety of data models, such as hypertext, graph-structured data, and relational data.

3.2 Requirements

A ranking procedure operating on collaboratively-edited datasets should exhibit the following properties:

¹ <http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/>

² <http://demo.openlinksw.com/rdfbrowser2/>

³ <http://www.w3.org/2005/ajar/tab>

⁴ <http://visinav.deri.org/>



Fig. 2. Rendering of information pertaining to the object <http://www.danbri.org/foaf.rdf#danbri> (datatype properties and values on the left, object properties and objects on the right, and data sources on the bottom). The decision on ordering of all data elements on the page has been taken based on rankings alone, without a priori schema knowledge.

- The use of an identifier owned by source s_a by a source s_b indicates an acknowledgement of the authority of s_a and should benefit the ranking of s_a
- Data providers who reuse identifiers from other sources should not be penalised, i.e. their data sources should not lose any rank value.
- A data provider who simply links to other important identifiers (requiring no external effort) should not gain any rank from doing so. Using only the node-link graph without taking into account the source (e.g. [4]) makes the ranking method receptive for spam: by adding a triple pointing from a popular URI to a spam URI, the spam URI gains rank from the popular URI.
- We cannot make any assumptions of directionality of links between objects, since link direction is arbitrary (is u_a related to u_b or u_b related to u_a ?). Thus we cannot use links occurring in the data graph as a vote.

3.3 Algorithm Outline

Our method for deriving a rank value for all data elements in a dataset consists of the following steps:

1. Based on the occurrence of identifiers $u \in S$, construct the naming authority graph $S \times S$ that serves as input to a fixpoint calculation.
2. The naming authority graph is used to derive PageRank scores for the data sources S .
3. The source ranks are used to derive a rank value for both global identifiers $u \in U$ and data elements with local scope $b \in B, l \in L$.

4 Naming Authority

The main point of ranking on the web is to rate popular pages higher than unpopular ones. Indeed, PageRank[15] interprets hyperlinks to other pages as votes. A possible adaptation for structured data sources would rank popular data sources higher than unpopular ones. However data models such as RDF do not specify explicit links to other web sites or data sources. Therefore a straightforward adaptation of PageRank for structured data is not possible, since the notion of a hyperlink (interpreted as a vote for a particular page) is missing.

However, a closer examination of the data model leads to the following observation: a crucial feature of structured data sources is the use of global identifiers. Typically, these global identifiers – URIs in case of the web – have a specified syntax, and exploit naming mechanisms such as the domain name system.

Consider Dan’s identifier `http://www.danbri.org/foaf.rdf#danbri`. Clearly the owner of the `danbri.org` domain can claim authority for creating this URI. Thus if the URI is used on `danbri.org` to denote Dan, the usage of the URI on other sites can be seen as a vote for the authority of the data source `danbri.org`.

To generalise this idea, one needs to define the notion of “naming authority” for identifiers. A naming authority is a data source with the power to define identifiers of a certain structure. Naming authority is an abstract term which could be applied to the provenance of a piece of information, be that a document, host, person, organisation or other entity. Data items which are denoted by unique identifiers may be reused by sources other than the naming authority.

We now define the general notion of naming authority:

Definition 1 (Naming Authority). *The naming authority of a global identifier $u \in U$ is the data source $s \in S$ which has the authority to mint the globally unique identifier u .*

4.1 Naming Authority for URIs

For naming authority in the RDF case, we assume a relation between $u \in U$ and $s \in S$ in the following way:

- if e contains a #, we assume the string before the # as the naming authority of the element, e.g. the naming authority for `http://danbri.org/foaf.rdf#danbri` is `http://www.danbri.org/foaf.rdf`.
- if e does not contain a #, we take the full element URI as the naming authority, e.g. the naming authority for `http://xmlns.com/foaf/0.1/maker` is `http://xmlns.com/foaf/0.1/maker`.

The Hypertext Transfer Protocol (HTTP)⁵, which is used to retrieve content on the web, allows the redirection of URIs, possibly multiple times, resulting in redirect chains. Redirect chains are unidirectional, i.e. the redirect relation does not follow the logical properties of the equivalence relation. To derive the correct naming authority we have to take HTTP redirects into account. Thus, for each naming authority, we check if the naming authority URI is redirected to another URI. If there is a redirect, we assume the redirected URI as the naming authority. E.g. <http://xmlns.com/foaf/0.1/maker> redirects to <http://xmlns.com/foaf/spec/>, hence the naming authority becomes <http://xmlns.com/foaf/spec/>. This is in-line with the Linked Data principles⁶.

4.2 Deriving the Naming Authority Matrix

As a first step, we derive the naming authority graph from the input dataset. That is, we construct a graph encoding links between data sources based on the implicit connections via identifiers.

Definition 2 (Naming Authority Matrix). *Given a data source $s_i \in S$ we define the naming authority matrix A as a square matrix defined as:*

$$a_{i,j} = \begin{cases} 1 & \text{if } s_i \text{ uses identifiers for which } s_j \text{ has naming authority} \\ 0 & \text{otherwise} \end{cases}$$

A naming authority graph for the example in Figure 1 is shown in Figure 3. In this example we assume the naming authority on a pay-level domain (PLD) level as described in the following.

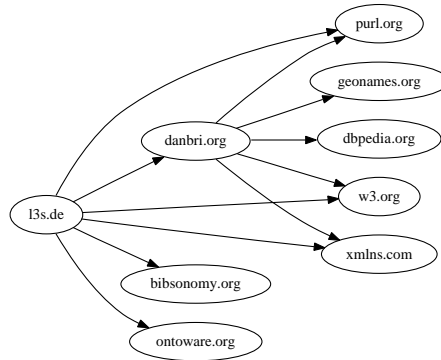


Fig. 3. Naming authority graph for data in Figure 1 on a pay-level domain granularity.

⁵ <http://www.ietf.org/rfc/rfc2616.txt>

⁶ <http://www.w3.org/DesignIssues/LinkedData.html>

4.3 Pay-Level Domains

In a variation of our algorithm we use the notion of pay-level domains (PLDs) as defined in [13]. A top-level domain (TLD) is a domain one level below the root in the DNS tree and appears as the label after the final dot in a domain name (e.g., `.com`, `.ie`). A pay-level domain (PLD) is a domain that must be paid for at a TLD registrar. PLDs may be one level below the corresponding TLD (e.g., `livejournal.com`), but there are many exceptions where they are lower in the hierarchy (e.g., `cam.ac.uk`).

Bharat and Henzinger [5] suggest using a host rather than a web page as the unit of voting power in their HITS-based [11] ranking approach. PLD-level granularity is preferable to domain or host-level granularity because some sites like Livejournal assign subdomains to each user, which would result in large tightly-knit communities if domains were used as naming authorities. The use of PLDs reduces the size of the input graph to the PageRank calculation.

Previous work has performed PageRank on levels other than the page level, for example at the more coarse granularity of directories, hosts and domains [10], and at a finer granularity such as logical blocks of text [6] within a page.

4.4 Internal vs External Links

Regardless of the level of granularity for naming authorities, there exist internal references occurring within a single naming authority, and external references occurring between different naming authorities. An internal reference is when a data source refers to an identifier which it has the authority to mint. An external reference is when a data source refers to an identifier which is under the authority of a different data source. Since our authority algorithm considers a reference as analogous to a vote for a source, it may be desirable to treat external references (from another source) differently to internal links (from the same source).

Similarly, in the HTML world, the HITS algorithm [11] considers only links which exist between different domains (which they call transverse links) and not links which occur within a single domain (which they call intrinsic links). The reason why only cross-domain links are taken into account in the HITS algorithm is that many intra-domain links are navigational and do not give much information about the authority of the target page. In variations of our algorithm we use either all links or take only external links into account.

5 Calculating Ranks

Having constructed the naming authority matrix, we now can compute scores for data sources, and in another step propagate data source scores to both global identifiers and identifiers and literals with a local scope which cannot be re-used in other sources. The algorithm can be implemented using a single scan over the dataset which derives both the naming authority matrix and a data structure that records the use of terms in data sources. As such, the method can be applied to streaming data. Subsequent calculations can then be carried out on the intermediate data structures without the use of the original data.

5.1 Calculating Source Ranks

For computing ranking scores we calculate PageRank over the naming authority graph. Essentially, we calculate the dominant eigenvector of the naming authority graph using the Power iteration while taking into account a damping factor.

In the input graph there may be sources which have no outlinks, referred to by the inventors of PageRank as dangling nodes [15]. The rank of these dangling nodes is split and distributed evenly across all remaining nodes. Conversely, there might be sources which have no inlinks, in the case where nobody uses the source’s identifier, or identifiers the source speaks authoritatively for; these sources only receive the damping factor plus the rank of the dangling nodes.

5.2 Calculating Identifier Ranks

Based on the rank values for the data sources, we now calculate the ranks for individual identifiers. The rank value of the individual identifier $u \in U$ depends on the rank values of the data sources $s \in S$ where the identifier occurs. The identifier rank of a global identifier $u \in U$ is defined as the sum of the ranks of the sources $s \in S$ in which u occurs.

$$identierrank(u) = \sum_{s \in \{s | u \in s; s \in S\}} sourcerank(s) \quad (1)$$

The identifier rank of local identifiers $b \in B$ and $l \in L$ are defined as the source rank of the source in which b or l occurs.

6 Experiments and Evaluation

In the following we evaluate our method on two real-world web datasets. We first introduce the datasets (one small and one large), then present runtime performance results, and finally present and analyse results of a quality evaluation of several variants of our method.

6.1 Datasets

We constructed two datasets:

- a small-scale RDF dataset crawled from a seed URI⁷. All unique URIs were extracted and their content downloaded in seven crawling rounds. The uncompressed dataset occupies 622MB of disk space.
- a large-scale RDF dataset derived from the 2008 Billion Triple Challenge datasets⁸. From these seed sets (around 450m RDF statements) we extracted all unique URIs and downloaded their contents during June and July 2008. The uncompressed dataset occupies 160GB of disk space.

Table 1 lists the properties of the datasets. In our experiments we followed one redirect, however, it is possible to take longer redirect chains into account.

⁷ <http://www.w3.org/People/Berners-Lee/card>

⁸ <http://challenge.semanticweb.org/>; we used the Falcon, Swoogle, Watson, SWSE-1, SWSE-2 and DBpedia datasets

Symbol	Small Dataset	Large Dataset
S	14k	6.5m
U	500k	74.3m
<i>tuple</i>	2.5m	1.1bn
Redirects	77k	4.5m

Table 1. Dataset properties.

6.2 Evaluation Variants

The experiments were carried out on five different algorithms which are enumerated in in Table 2. The methods evaluated include four variants of our algorithm which differ according to the level of the naming authority (URI or PLD), and the references which we took into consideration for the authority calculation (all references, or external references only). We compared our method to a naive version of PageRank operating directly on the node-link graph without taking sources into account. We did not compare to ObjectRank [4] because ObjectRank requires manual assignment of weights to each of the thousands of properties in the dataset, which is infeasible.

On the small dataset, we compared performance of all the methods listed in Table 2, and carried out a quality evaluation on the results. On the large dataset, we conducted a scale-up experiment on the EU variant to demonstrate the scalability of our algorithm. We performed all experiments on a quad-core Xeon 2.33GHz machine with 8GB of main memory and a 500GB SATA drive.

Method	Description
AU	All links contribute authority URI-level naming authorities
EU	External links exclusively contribute authority URI-level naming authorities
AP	All links contribute authority PLD-level naming authorities
EP	External links exclusively contribute authority PLD-level naming authorities
PR	PageRank over the object graph

Table 2. Ranking methods evaluated.

The implementation assures the uniqueness of links in the naming authority graph via sorting, and aggregates rank scores by writing the rank fractions to a file, sorting the file to group common identifiers together, and summing up the values via file scan. For the PageRank calculation, we fixed the number of iterations to ten rather than having the method converge in specified error bounds, which means we have to maintain only the current version of the rank vector rather than also maintaining the version from the previous iteration for comparison.

6.3 Performance Evaluation

The runtime of the five tested variants on the small dataset is plotted in Figure 4. Each processing step is plotted separately. The relatively large amounts of time spent on the AP and EP variants is due to the reduction of full URIs to pay-level domains. Given the coarser granularity of pay-level domains, the derived naming authority graph is quite small and thus accounts for the short running time of the PageRank calculation. In all approaches, a significant time is spent on the pre-processing of the data, processing steps which could be either carried out in-memory for small datasets or could be distributed across machines if required.

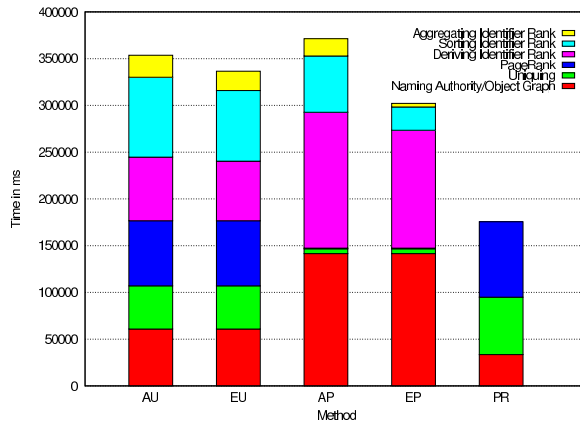


Fig. 4. Runtime of algorithm variations.

6.4 Scale-Up Experiment

We also conducted a scale-up experiment in which the algorithm performed ranking on a dataset with 1.1bn statements. In the scale-up experiment we used the Linux command `sort` with 6GB of main memory for sorting and unquing instead of our standard Java implementation. The runtime of each processing step is listed in Table 3.

Processing Step	Duration
Deriving Naming Authority Graph	55h36m22s
Uniquing Naming Authority Graph	40h17m52s
PageRank Calculation	12h30m24s
Deriving Identifier Ranks	39h7m13s
Sorting Identifier Ranks	14h36m27s
Aggregating Identifier Ranks	1h56m43s

Table 3. Runtime of processing steps for the large dataset.

6.5 Quality Evaluation

In the information retrieval community, there are clearly defined processes and well-established metrics for evaluating how well an system performs in meeting the information requirements of its users. Standard test collections consisting of documents, queries, and relevance judgements are available and are widely used. Search over Semantic Web data is a relatively new area however, and equivalent labelled collections do not yet exist. Therefore, given the lack of a labelled dataset, we use an alternative approach to quality assessment.

To compare the quality of the methods, we conducted a study in which we asked participants to manually rate results of queries for each algorithm. For every query, we presented the evaluators with five different ranked lists, each corresponding to one of the ranking methods. The result lists consisted of the top ten results returned by the respective method.

The evaluators were asked to order these lists from 1 to 5, according to which lists they deemed to represent the best results for each query. Our analysis covered the five scenarios listed in Table 4. For each scenario there were between 11 and 15 evaluators. Scenarios 1 and 2 were evaluated by the participants of a Semantic Web related workshop held in November 2008. During this evaluation, we presented each participant with results to a general query (S1) and with results of a query for their own name (S2), which was possible since all eleven participants have FOAF files and thus satisfactory data coverage. Scenarios 3 - 5 were evaluated by Ph.D. students and post-docs in the university. These final three scenarios were queries for people with whom the evaluators are familiar.

Scenario	Request	N
S1	Query for all persons in the dataset	11
S2	Keyword search: evaluator's own name	11
S3	Keyword search: "Tim Berners-Lee"	15
S4	Keyword search: "Dan Brickley"	15
S5	Keyword search: "John Breslin"	15

Table 4. Scenarios used for evaluation. N is the number of evaluators.

For each scenario we plotted the mean rank assigned by evaluators for each method. Error bars represent one standard deviation. We determine significance of the results using the Wilcoxon test with $p < .05$.

Figure 5 shows the average ranks which resulted for S1, a query for all people in the dataset. For this query, the top ten results of the methods AU and EU were identical. The differences between methods in this table are all statistically significant. The evaluators were almost unanimous in ranking this particular query which is why for three of the points there is no standard deviation marked.

Figure 6 shows the average ranks which were assigned by evaluators to a query for their own name (S2). In this table, the differences between each variant of our method and the PageRank method are all statistically significant. However the differences between the variants of our method are not statistically significant.

Figures 7, 8 and 9 show the average ranks which resulted for queries for three people involved in the Semantic Web community - scenarios S3, S4 and S5. The differences between each variant of our method and the PageRank method are statistically significant for all queries, with one exception (scenario S4, methods AP and OR). For S5, every evaluator rated PR as the worst method so for the corresponding point there is no standard deviation marked. The differences between the variants of our method are generally not statistically significant.

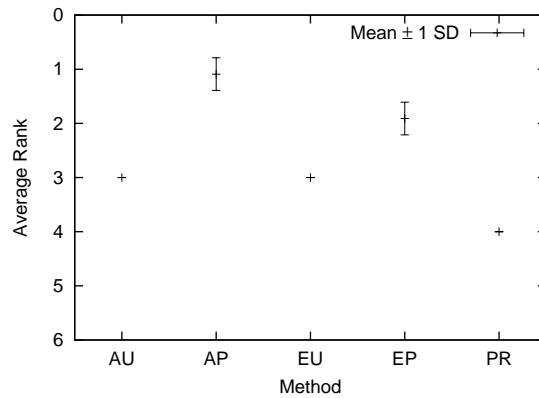


Fig. 5. Average ranks for scenario S1: Query for all persons in the dataset.

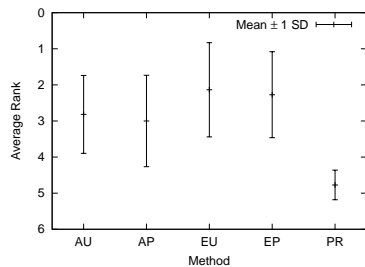


Fig. 6. Average ranks for scenario S2: Keyword search for the evaluator’s own name.

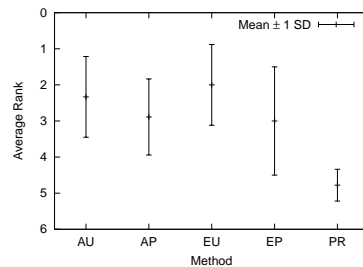


Fig. 7. Average ranks for scenario S3: Keyword search for “Tim Berners-Lee”.

For scenarios 2 - 5 the average ranks follow similar patterns, showing that the evaluator’s assessments of the methods were consistent over different queries.

We can conclude from the quality evaluation that our algorithm gives significantly better results than simply implementing PageRank on the object graph.

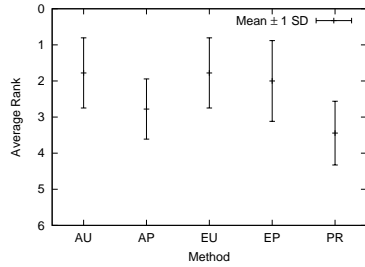


Fig. 8. Average ranks for scenario S4: Keyword search for “Dan Brickley”.

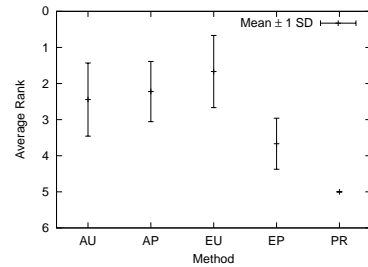


Fig. 9. Average ranks for scenario S5: Keyword search for “John Breslin”.

We cannot determine the best variant of our algorithm with statistical significance. However with the exception of the query for all people in the dataset, the best method is always EU (where naming authorities are URIs, and only external links contribute to authority).

7 Related Work

Citation-based algorithms have been investigated in sociology in the area of social network analysis [16], which states as unresolved issue the mismatch between two-dimensional graph theory and multi-dimensional social networks. We have extended links-based connectivity algorithms such as PageRank [15] and HITS [11] to operate on data graphs. PageRank scales very well but only operates on two-dimensional matrices, the graph derived from the hyperlink structure.

An extension to PageRank and HITS to the multi-dimensional case is TOPHITS [12], a ranking procedure rooted in multilinear algebra which encodes the hyper-text link structure (including link labels) as a three-way tensor. Rather, our approach operates on a general model for semistructured data, and is centred around the notion of trustworthiness of data sources.

ObjectRank [4] is an approach to rank a directed labelled graph using PageRank. The work includes a concept called authority transfer schema graphs, which defines weightings for the transfer of propagation through different types of links. ObjectRank relies on user input to weight the connections between nodes to describe their semantic weight, so that the three-way representation can be collapsed into a two-way matrix, on which a PageRank-style algorithm is applied. Our approach does not require any manual input, which is not feasible given the scale and heterogeneity of the input. In addition, omitting the provenance of data as in ObjectRank opens up the method to abuse - anyone could maliciously link to their own identifiers from well-known, highly ranked identifiers and therefore gain reputation by association. Using our notion of naming authority, reusing popular identifiers only results in a propagation of reputation from the containing sources to the popular source. As an added benefit, taking into account the naming authority results in a much smaller graph for PageRank calculation.

ReConRank [8] applies a PageRank-type algorithm to a graph which unifies the documents and resources in a dataset. The method generates scores for the documents and entities in a collection, but not for the properties. ReConRank does take data provenance into account, however because it simultaneously operates on the object graph, it is still susceptible to spamming.

SemRank [3] ranks relations and paths on Semantic Web data using information-theoretic measures. In contrast, we assign a rank value to all identifiers occurring in the data sources, based on a fixpoint calculation on the naming authority graph.

Swoogle [7] ranks documents using the OntoRank method, a variation on PageRank which iteratively calculates ranks for documents based on references to terms (classes and properties) defined in other documents. We extend the method described in [7] in several important ways: we generalise the notion of term use to naming authority which establishes a connection between identifier and source; we include the PLD abstraction layer which has been found to be advantageous for ranking in the web environment; and we extend our ranking scheme to not only cover vocabulary terms but instance identifiers as well, which is important in our Linked Data browser use-case scenario.

The notion of naming authority is related to that of authoritative sources as considered by the SAOR reasoning system [9]. SAOR uses authoritative sources to determine whether a source has authority to extend a class or property, while we use naming authority to rank sources and identifiers.

AKTiveRank [1] is a system for ranking ontologies based on how well they cover specified search terms. AKTiveRank combines the results of multiple analytic methods to rank each ontology. Individual instances and vocabulary terms are not ranked. Ontocopi [2] provides a way of locating instances in a knowledge base which are most closely related to a target instance. The Ontocopi tool uses a spreading activation algorithm and allows both manual and automatic tuning. However the source of data is not taken into consideration. Similarly, the Sem-Search system [14] ranks entities according to how well they match the user query but does not consider the source of data.

8 Conclusion

Ranking provides an important mechanism to prioritise data elements and assuage the noise inherent in datasets which have been aggregated from disparate sources or have been created in a decentralised way. We have demonstrated a set of scalable algorithms for ranking over a general model of structured data collected from an open, distributed environment, based on the notion of naming authority. We adapted the general model to the case of RDF, taking the intricacies of RDF data from the web into account.

In comparison to using plain PageRank on a node-link graph representation of RDF, our methods exhibit similar runtime properties while improving on the quality of the calculated rankings. Contrary to methods which require manual input of a domain expert to specify schema weights, our method derives rankings for all identifiers in the dataset automatically.

We foresee our method having applications in search, query processing, reasoning, and user interfaces over integrated datasets from a large number of sources,

an environment where assessing trustworthiness of sources and prioritising data items without a priori schema knowledge is vital.

References

1. H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with AKTiveRank. In *5th International Semantic Web Conference*, pages 1–15, 2006.
2. H. Alani, S. Dasmahapatra, K. O’Hara, and N. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, 2003.
3. K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *14th International Conference on World Wide Web*, pages 117–127, 2005.
4. A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: authority-based keyword search in databases. In *Proceedings of the 13th International Conference on Very Large Data Bases*, pages 564–575, 2004.
5. K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, 1998.
6. D. Cai, X. He, J. Wen, and W. Ma. Block-level link analysis. In *27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 440–447, 2004.
7. L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *4th International Semantic Web Conference*, pages 156–170, 2005.
8. A. Hogan, A. Harth, and S. Decker. ReConRank: A scalable ranking method for semantic web data with context. In *2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.
9. A. Hogan, A. Harth, and A. Polleres. SAOR: Authoritative Reasoning for the Web. In *3rd Asian Semantic Web Conference*, pages 76–90, 2008.
10. X.-M. Jiang, G.-R. Xue, W.-G. Song, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Exploiting PageRank at Different Block Level . In *5th International Conference on Web Information Systems*, pages 241–252, 2004.
11. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
12. T. Kolda, B. Bader, and J. Kenny. Higher-order web link analysis using multilinear algebra. In *5th IEEE International Conference on Data Mining*, pages 242–249, 2005.
13. H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov. Irlbot: scaling to 6 billion pages and beyond. In *17th International Conference on World Wide Web*, pages 427–436, 2008.
14. Y. Lei, V. Uren, and E. Motta. Semsearch: A search engine for the semantic web. In *14th International Conference on Knowledge Engineering and Knowledge Management*, pages 238–245, 2006.
15. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
16. J. Scott. Trend report: Social network analysis. *Sociology*, 22(1):109–27, 1988.