DING! Dataset Ranking using Formal Descriptions

Nickolai Toupikov, Jürgen Umbrich,
Renaud Delbru, Michael Hausenblas,
Giovanni Tummarello
DERI, National University of Ireland, Galway,
IDA Business Park, Galway, Ireland
firstname.lastname@deri.org

ABSTRACT

Considering that thousands if not millions of linked datasets will be published soon, we motivate in this paper the need for an efficient and effective way to rank interlinked datasets based on formal descriptions of their characteristics. We propose DING (from **D**ataset Rank**ING**) as a new approach to rank linked datasets using information provided by the voiD vocabulary. DING is a domain-independent link analysis that measures the popularity of datasets by considering the cardinality and types of the relationships. We propose also a methodology to automatically assign weights to link types. We evaluate the proposed ranking algorithm against other well known ones, such as PageRank or HITS, using synthetic voiD descriptions. Early results show that DING performs better than the standard Web ranking algorithms.

1. MOTIVATION

Following Marshall and Shipman [1], we understand linked datasets in terms of the distributed database perspective. The primary targeted consumers are expected to be machines; a fair degree of automation needs to be guaranteed in order to enable new types of Web applications. While nowadays the number of datasets—published in accordance to the linked data principles [2]—is somewhat limited, this is expected to change soon. Considering thousands if not millions of linked datasets[1], one can expect to get lost, soon when trying to identify appropriate datasets for a certain task.

Two issues come to mind when talking about selecting (possibly many) datasets: **efficiency** and **effectiveness**. While the former basically refers to how fast certain datasets can be identified, the latter focuses on the relevancy, that is how well the dataset fulfills the stated requirements in a certain context (the domain of the Web application). When faced with a list of potential candidates, one usually wants to rank them according to certain criteria in order to select the most relevant ones.

Our thesis at hand now is that, based on a formal (high-level) description of a dataset's content and interlinking provided by voiD, Semantic Web clients can effectively rank datasets using well-known strategies such as PageRank [4] or HITS [5] in a very efficient way. Without such high-level descriptions, the client would have to "crawl" a large number of documents in order to analyze and derive precise statistics about the content of a dataset, hereby requiring an excessive amount of time and resources.

The rest of the paper is structured as follows: the next section discusses exitsing approaches. Then, we lay out the foundations regarding the formal description of linked datasets in sec. 3 and render our proposal in detail (sec. 4). Further, in sec. 5, we report on early findings when comparing our approach to widely used ones such as PageRank or HITS. We conclude in sec. 6 by discussing the proposed ranking methodology and point out possible future steps.

2. EXISTING WORKS

Link analysis has proven to be effective for query independent quality web search. PageRanks [4] and HITS [5] have been successfully applied to measure the importance of web pages by analysing their link structure. These two algorithms consider only one type of links, i.e. hyperlinks, but has been shown to improve the effectiveness of web search engines [6, 7].

When working on a finer granularity level - such as entity level - with more heterogeneous links, the previous approaches are no longer applicable. In such condition, by assuming that links are equivalent, the analysis of entity relationships does not provide accurate results since links of different types can have various impact on the ranking computation.

Recent works [8, 9] have extended PageRank to consider different types of relations between entities or objects. PopRank [10], a domain independent object-level link analysis, proposes a machine learning approach to automatically assigns a "popularity propagation factor" to each type of relations. ObjectRank [11] goes further by applying authority-based ranking to keyword search in databases where various objects are connected with semantic relations.

The Swoogle search engine [12] was the first one to propose, *OntoRank*, an adaptation of PageRank for Semantic Web resources. In their work, they compute popularity of resources based on three levels of granularity: documents, terms and RDF graphs. In [13], a link analysis is applied at query time for computing the popularity of resources and contexts (which can be seen as documents or datasets).

[1] A simple estimation might support this argument: take for example relational databases such as MySQL found in nearly every modern Web application, or the manifold repositories in the software development domain (for example, CVS or SVN) or registries (LDAP, OPACs, etc.)—each of them, once on the Web of Data (using out-of-the-box linked data publishing tools such as Triplify [3]) represents at least one dataset.

LDOW2009, April 20, 2009, Madrid, Spain
.

Their approach differentiates two levels of link analysis, resources and context graphs, and the different relationships between them.

In this paper, we are studying how to improve search results by ranking datasets according to their popularity. Our approach is based on link analysis between datasets by using the information provided by the voiD descriptions. We consider the types of relationships but also the cardinality of link sets. We propose also an automatic weighting scheme to find appropriate weights for relation types.

3. DESCRIBING DATASETS

In order to realise our vision of a semantic ranking, we build upon a formal description of the datasets and their interlinking. Only recently the *Vocabulary of Interlinked Datasets* (voiD) [14] has been released; voiD is an RDFS vocabulary for describing linked datasets. A dataset in voiD 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". Interlinking in voiD is modeled utilising a so called linksets. A linkset in voiD is "a subset of a dataset, used for storing triples to express the interlinking relationship between datasets; in each interlinking triple, the subject is a resource hosted in one dataset and the object is a resource hosted in another dataset".

Given that such voiD descriptions are published alongside with the datasets, they can be collected via pings, by crawling, or simply follow-your-nose by a semantic indexer such as Sindice [15] or the Yahoo! Search Monkey [16]. We assume such a collection of voiD descriptions in the following. We note further that, as voiD being metadata about linked data, is RDF-grounded, we can use all current RDF tools and libraries to process, store and visualise it. Further, it is perfectly possible to go from the meta-level to the meta-meta-level, that is having a voiD description `about` voiD descriptions.

4. DING—DATASET RANKING

Our proposal for a semantic ranking of RDF datasets is called DING (from **D**ataset Rank**ING**) and is based on voiD descriptions of the datasets.

4.1 Exploiting voiD's characteristics

Based on the voiD guide [17] we will review the relevant features of voiD in the following and discuss their suitability with respect to dataset selection and ranking.

- The **size of the dataset**, that is, for example the number of triples or the number of distinct subjects can be used for ranking. In voiD this is a `void:statItem` property along with one of five predefined dimensions such as `void:numberOfTriples` or `void:numberOfDocuments`. We have argued in [18] recently that the sheer numbers of triples is likely not a good measure for its value.

- **Categorisation** of datasets in voiD is done using `dcterms:subject` along with DBpedia [19] resources. This can be used in a first step to massively decrease the search space. It acts as a sort of lexicon allowing to lookup a category and find related datasets. As a second step, DING can be used to rank the list of datasets matching a certain category.

- The **interlinking** of a dataset in voiD, that is, its outgoing and incoming links, is represented using the `void:linkPredicate` property. We identify two potential dimensions that might be exploited for ranking:
  - regarding the semantics of the links (such as `rdfs:seeAlso` vs. `foaf:knows`) and
  - on a quantitative level, that is regarding the number of interlinking triples.

- The kind of and number of used vocabularies in a dataset can be seen from the `void:vocabulary` property value.

- Other voiD characteristics such as `void:uriRegexPattern` or the technical features of a dataset (such as available serialisations) via `void:TechnicalFeature`) can not directly be used for ranking, though perfectly for filtering (as in case with categorisation).

The following example in Fig. 1 may help highlight our thinking:

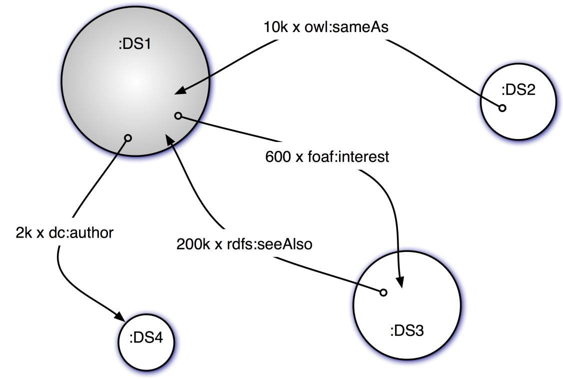

Figure 1: Exemplary collection of four voiD descriptions.

4.2 DING! Implementation

In this section, we present how we adapted the weighted PageRank algorithm in order to perform the Dataset ranking based on their interconnection. We then explain how it is possible to assign automatically a weight to a certain link type.

PageRank is a ranking system that originates from Web Search Engines using a random walk algorithm. The Ranking system evaluates the probability of finding the web surfer on any given page. This algorithm is based on the assumption that when someone publishes a resource on the web, he will do his best to link the published resource — be it a web page, or in our case — a dataset — to the most relevant and trustworthy resources availiable on the web. Hence the relevancy is assumed to be related to a high degree of inlinks from other web resources. And from a probabilistic point of view — the more inlinks a dataset has, the most likely the 'random surfer' will be lead to it in his journey.

The original PageRank $r(P_i)$ of a web page i is given by

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \qquad (1)$$

```
:DS1 a void:Dataset ;
     foaf:homepage <http://example.org/cats/> ;
     dcterms:subject
          <http://dbpedia.org/resource/Cats> ;
     void:subset :DS1toDS3  ;
     void:subset :DS1toDS4  .

:DS2 a void:Dataset ;
     foaf:homepage <http://petfood.example.org/> ;
     dcterms:subject
          <http://dbpedia.org/resource/Cats> ;
     dcterms:subject
          <http://dbpedia.org/resource/Pet_foods> ;
     void:subset :DS2toDS1  .

:DS1toDS3 a void:Linkset ;
          void:subjectsTarget :DS1 ;
          void:objectsTarget :DS3 ;
          void:linkPredicate foaf:interest ;
          void:statItem [
              rdf:value 600;
              scovo:dimension void:numberOfTriples;
          ] .

:DS2toDS1 a void:Linkset ;
          void:target :DS1 ;
          void:target :DS2 ;
          void:linkPredicate owl:sameAs ;
          void:statItem [
              rdf:value 10000;
              scovo:dimension void:numberOfTriples;
          ] .
```

Listing 1: An exemplary voiD description.

Where B_{P_i} is the set of pages linking to P_i and $|P_j|$ is the total number of pages linked by P_j . Hence, $\frac{1}{|P_j|}$ is in fact the probability for the random surfer to choose to go from P_j to P_i out of all pages linked by P_j. This probability referred to as $p_{j\to i}$, can be modified in order to provide a weighting of the "importance" of the hyperlink.

The parallel from web documents to voiD descriptions is done in a naive way. The web pages are now datasets, and the hyperlinks correspond to linksets joining the dataset they belong to another:

- P_i corresponds to an element D_i define by `void:dataset`.
- A hyperlink form in the page P_i pointing to the page P_j will correspond to a `void:linkset` element connecting D_i and D_j, defined as `void:subset` of a dataset D_i. The linkset will be referred to as $L_{i\to j}$. We also define $n(L_{i\to j})$ as the number of relations in the linkset, and $s(L_{i\to j})$ as the predicate declared in the linkset. For example, in Fig. 1 $n(L_{1\to 3}) = 600$ and $s(L_{1\to 3}) =$ `"foaf:interest"`. $\mathcal{L}$ is the set of linksets defined in the entire data collection.
- Similarly to the set B_{P_i} of pages linking to P_i, we define $O(i) = \{j | \exists L_{i\to j} \in \mathcal{L}\}$ as the indices of datasets linked from D_i.
- $p_{i\to j}$ can be modified according to the information available about the linkest $L_{i\to j}$, such as $s(L_{i\to j})$ or $n(L_{i\to j})$, as well as general statistics over $\mathcal{L}$.

Like in the web page link analysis, the links between datasets deserve a deeper analysis in order to obtain a finer ranking. For example in Fig. 1 the probability of the user going from DS1 to DS3 is likely to be different from the probability of going to DS4 - since the predicate and number of links associated to $L_{1\to 3}$ are not the same as the ones associated to $L_{1\to 4}$.

The goal will hence be to define a weight function $w(L_{i\to j})$. The weight will then be normalized in order to generate the transition probability $p_{i\to j}$ as follows.

$$p_{i\to j} = \frac{w(L_{i\to j})}{\sum_{k\in O(i)} w(L_{i\to k})} \tag{2}$$

The first approach is simply to define $w(L_{i\to j}) = n(L_{i\to j})$. In the case of Fig. 1, $p_{1\to 3} = \frac{600}{2000+600} \simeq 0.23$ and $p_{1\to 4} = \frac{2000}{2000+600} \simeq 0.77$. However, this definition does not take into account the nature of the link, and the likelihood that the user may well chose foaf:interest above dc:author to browse into another dataset. As a result, additional weights can be assigned based on the nature of the predicate involved in the link.

The values assigned can be either statically predefined, or computed dynamically, given the accumulated voiD information. We present our approach, based on TF-IDF a well known algorithm when it comes to weight the relevance of a term(in our case - the predicate), given its frequency in a data collection. Hence, the weight, given by TF-IDF would be

$$TF(L_{i\to j}) = \frac{n(L_{i\to j})}{\max_{k\in O(i)} n(L_{i\to k})} \tag{3}$$

$$IDF(s(L_{i\to j})) = \log \frac{N}{1 + freq(s(L_{i\to j}))} \tag{4}$$

Where $freq(s(L_{i\to j}))$ is the frequency of occurrence of linksets using the predicate of $L_{i\to j}$ in the collection's datasets. Finally, we define w as

$$w(L_{i\to j}) = TF(L_{i\to j}) \times IDF(s(L_{i\to j})) \tag{5}$$

5. EXPERIMENTS AND EARLY FINDINGS

In order to verify our thesis that formal descriptions of linked datasets help yielding better results for the ranking of the datasets, we have set up an evaluation framework that executes various ranking algorithms on a synthetic voiD description[2](see Fig. 2). It is composed of 15 artificial dataset descriptions interlinked using 8 different predicates and partitioned into two clouds (datasets 1 to 9 and 10 to 15). The experiment used several ranking algorithms to estimate the generic relevancy of every artificial dataset within the synthetic cloud.

5.1 The setup

For the evaluation we use the Java Universal Network/-Graph Framework (JUNG)[3] to compare the DING algorithm with other established and well known ranking algorithms. Further, we use a naive link-sum rank function (*DRank*) as a baseline to discuss the results. Three out of the four ranking algorithms are also extended with the DING link weight function. In detail we evaluate and compare the following ranking algorithms:

[2] The full benchmark data is available at `http://sw.deri.org/2009/02/DING/example-void-collection.ttl`. Unfortunately no real-world voiD cloud was readily availiable for the experiment at the time of writing.

[3] http://jung.sourceforge.net/

Rank	1	2	3	4	5	6
STD PageRank	DS1 (0.20)	DS3 (0.15)	DS4 (0.14)	DS11 (0.12)	DS13 (0.072)	DS10 (0.056)
DING PageRank	DS4 (0.18)	DS1 (0.14)	DS11 (0.12)	DS13 (0.091)	DS3 (0.081)	DS10 (0.074)
DRank	DS10 (0.35)	DS13 (0.16)	DS12 (0.16)	DS11 (0.12)	DS15 (0.11)	DS14 (0.044)
HITS	DS1 (0.43)	DS4 (0.28)	DS2 (0.11)	DS3 (0.094)	DS11 (0.022)	DS10 (0.017)

Table 1: Evaluation results: the top 6 datasets for each ranking algorithm with their normalized score

- **DRank**: A baseline ranking algorithm using a naive approach. The datasets are ranked according to the number of links they have with other datasets.
- **PageRank** Google's page rank algorithm [4].
- **DING PageRank** modification of the PageRank Algorithm as described in Sec 4.2
- **HITS** Another well known ranking algorithm is HITS [5]. For each data set in the voiD graph a "hubs-and-authorities" importance measure is calculated.

5.2 Results

Table 1 lists the results of the evaluation. The naive ranking approach, *DRank*, completely leaves out the first cloud for having much less links in its linksets than the second cloud. We see that standard PageRank and HITS algorithms do not take into account the nature of the links and rank DS1 first. Although DS1 is indeed heavily linked by other datasets, it is mostly inlinked by "weak" links like `owl:sameAs` or `rdfs:seeAlso`. The information-theory view defined in tfidf suggests that these links — being the most common ones — do not hold as much information content as less common ones, and are therefore less significant. For example while looking for information about an article, the user will get more precise information following `dcterms:author` than a generic property such as `rdfs:seeAlso`, and is hence more likely to follow the former. As a result a dataset linked by uncommon links will likely be more significant than one linked by common ones - and should have a higher voiD ranking.

Another advantage of PageRank that makes it very relevant for the Linked Data approach is that it gives a low ranking to datasets that do not have inlinks. The value of a dataset within the cloud is dependent on how well it is linked by other datasets.

6. CONCLUSION

We have presented DING, a new approach to rank linked datasets based on voiD descriptions. Though one might object that currently there are not many voiD descriptions available[4] we argued that this is very likely to change soon. Further, the infrastructure to collect voiD descriptions is in place (voiD being RDF, the requirements to do so are minimal).

We have motivated the need for a efficient and effective way to rank datasets based on their characteristics (content-wise and with respect to the interlinking). Finally we have shown how DING performs in relation to existing ranking algorithms and discussed the results.

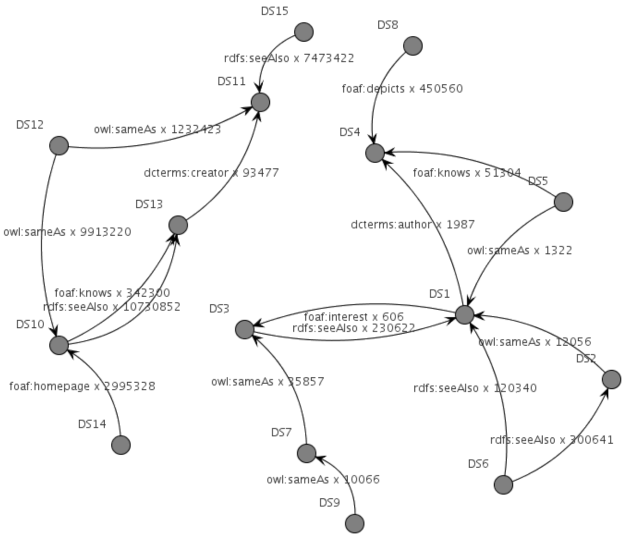

Figure 2: Visualisation of the synthetical dataset network

Acknowledgements

Our work has partly been supported by the European Commission under Grant No. 217031, FP7/ICT-2007.1.2, project "Domain Driven Design and Mashup Oriented Development based on Open Source Java Metaframework for Pragmatic, Reliable and Secure Web Development" (Romulus)[5], by the European FP7 project *Okkam - Enabling a Web of Entities* (contract no. ICT-215032), and by Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

7. REFERENCES

[1] C.C. Marshall and F.M. Shipman. Which Semantic Web? In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 57–66, New York, NY, USA, 2003. ACM.

[2] T. Berners-Lee. Linked Data. `http://www.w3.org/DesignIssues/LinkedData.html`, 2007.

[3] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify - Lightweight Linked Data Publication from Relational Databases. In *International World Wide Web Conference (WWW 09), Madrid, Spain*, page upcoming, 2009.

[4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking:

[4] Indeed one finds voiD descriptions at time of writing, already; see for example `http://void.rkbexplorer.com/`.

[5] `http://www.ict-romulus.eu/`

Bringing order to the web. Technical Report 1999-66, November 1999.

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

[6] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[7] Soumen Chakrabarti, Byron E. Dom, David Gibson, Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Mining the web's link structure. *Computer*, 32(8):60–67, 1999.

[8] Wenpu Xing and Ali Ghorbani. Weighted pagerank algorithm. In *CNSR '04: Proceedings of the Second Annual Conference on Communication Networks and Services Research*, volume 0, pages 305–314, Washington, DC, USA, 2004. IEEE Computer Society.

[9] Ricardo Baeza-Yates and Emilio Davis. Web page ranking using link attributes. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 328–329, New York, NY, USA, 2004. ACM.

[10] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen, and Wei-Ying Ma. Object-level ranking: bringing order to Web objects. In *Proceedings of the 14th international conference on World Wide Web - WWW 05 WWW 05*, page 567. ACM, 2005.

[11] Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou. Objectrank: authority-based keyword search in databases. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 564–575. VLDB Endowment, 2004.

[12] Li Ding, Rong Pan, Timothy W. Finin, Anupam Joshi, Yun Peng, and Pranam Kolari. Finding and ranking knowledge on the semantic web. In *International Semantic Web Conference*, pages 156–170, 2005.

[13] Aidan Hogan, Andreas Harth, and Stefan Decker. Reconrank: A scalable ranking method for semantic web data with context. In *Proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006)*, Athens, GA, USA., 11 2006.

[14] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009.

[15] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the Open Linked Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pages 552–565, 2007.

[16] P. Mika. Microsearch: An Interface for Semantic Search. In *Semantic Search, International Workshop located at the 5th European Semamntic Web Conference (ESWC 2008)*, volume 334 of *CEUR Workshop Proceedings*, pages 79–88. CEUR-WS.org, 2008.

[17] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. voiD guide—Using the Vocabulary of Interlinked Datasets. Community Draft, voiD working group, 2009.

[18] Michael Hausenblas, Wolfgang Halb, Yves Raimond, and Tom Heath. What is the Size of the Semantic Web. In *Proceedings of I-Semantics 2008, Graz, Austria*, 2008.

[19] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pages 722–735, 2007.