Hierarchical Link Analysis for Ranking Web Data

Renaud Delbru[1], Nickolai Toupikov[1], Michele Catasta[2]*, Giovanni Tummarello[1,3], and Stefan Decker[1]

[1] Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
`firstname.lastname@deri.org`
[2] School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
`firstname.lastname@epfl.ch`
[3] Fondazione Bruno Kessler
Trento, Italy
`lastname@fbk.eu`

Abstract. On the Web of Data, entities are often interconnected in a way similar to web documents. Previous works have shown how PageRank can be adapted to achieve entity ranking. In this paper, we propose to exploit locality on the Web of Data by taking a layered approach, similar to hierarchical PageRank approaches. We provide justifications for a two-layer model of the Web of Data, and introduce DING (Dataset Ranking) a novel ranking methodology based on this two-layer model. DING uses links between datasets to compute dataset ranks and combines the resulting values with semantic-dependent entity ranking strategies. We quantify the effectiveness of the approach with other link-based algorithms on large datasets coming from the Sindice search engine. The evaluation which includes a user study indicates that the resulting rank is better than the other approaches. Also, the resulting algorithm is shown to have desirable computational properties such as parallelisation.

1 Introduction

A growing number of data management scenarios have to deal with heterogeneous and inter-linked data sources. On the Web, more and more structured and semi-structured data sources are becoming available: millions of databases supporting simple web applications or more advanced Web 2.0 services (e.g. Wordpress, Facebook), hundreds of millions of documents embedding semi-structured data (e.g. HTML tables, Microformats such as Last.fm, RDFa such as Best-Buy, etc.), and recently the rapidly growing amount of online Semantic Web

* The author contributed to this work while he was a master student in DERI

data repositories (e.g. the Linked Data[4] initiative). In enterprises, integration of structured data (databases) and semi-structured data (XML, documents, etc.) is a common scenario. There is a growing demand for the exploitation of these data sources, and therefore a need for searching and retrieval.

These inter-linked datasets constitute the Web of Data. The content of every dataset can be transposed into a graph model, representing entities (i.e. information resources) and their relationships. Compared to the Web of Documents, the unit of information is of smaller granularity. As a consequence the number of nodes and links is orders of magnitude larger than on the Web of Documents. As the Web of Data graph is very large, containing billions of nodes and edges, developing scalable link analysis algorithm for computing popularity score on web-scale data graph is becoming an important requirement.

Current link analysis algorithms [1, 2] for the Web of Data consider exclusively the graph of entities. In addition to their high computational complexity, they suffer from not taking into account the semantics of datasets which produce sub-optimal results. For example, given a dataset about employees, one would like to find the most skilled employee and not the most popular one. In this paper, we introduce a two-layer model for the Web of Data and provide justifications for this two-layer model. Based on this model, we propose a novel ranking algorithm called DING (for Dataset rankING). DING performs ranking in three steps: 1. dataset ranks are computed by performing link analysis on the top layer (i.e. the dataset graph); 2. for each dataset, entity ranks are computed by performing link analysis on the local entity collection; 3. the popularity of the dataset is propagated to its entities and combined with their local ranks to estimate a global entity rank.

2 Background

In this section, we introduce a model of the Web of Data that will serve as framework in the rest of the paper. We then show how the original PageRank [3] algorithm can be adapted to this model.

2.1 Web Data Model

For the purpose of this paper, we need a generic graph model that encompasses the different use cases discussed previously. Therefore, we define a labelled directed graph model that covers the various data sources found on the Web of Data, i.e. Microformats, RDFa, Semantic Web repositories, etc. This graph model represents entities and their relationships. We denote by entity a self-contained unit of information that has relationships with other entities. Typical examples of entities include documents, persons, events, products, etc.

Let U be a set of node labels, and V a set of edge labels. The Web of Data is defined as a graph over U and V, and is a tuple $G = \langle E, L, \lambda \rangle$ where E is a set of

[4] Linked Data: `http://linkeddata.org/`

nodes representing entities, $L \subseteq \{(e_1, \sigma, e_2) | e_1, e_2 \in E, \sigma \in V\}$ a set of labelled edges representing the relationships (or links) and $\lambda : E \to U$ a node labelling function. The components of an edge $l \in L$ will be denoted by $source(l)$, $label(l)$ and $target(l)$ respectively.

Let a dataset D be a subgraph of G. A dataset D is a tuple $D = \langle E_D, L_D, \lambda, \Delta_D \rangle$ with $E_D \subseteq E$ and $L_D \subseteq L$. Two datasets are not mutually exclusive and their nodes may overlap, i.e. $E_{D_1} \cap E_{D_2} \neq \emptyset$. We identify a subset $\Delta_D \subseteq U$ as a set of internal node labels to a dataset D, i.e. the set of entity identifiers that originates from this dataset. For example, such a set could be the URIs defined by the naming authority of the dataset [4]. A node e of a graph D is said to be *internal* if $\lambda(e) \in \Delta_D$, otherwise it is said to be *external* (i.e. it identifies an entity from another dataset). Analogously, an edge l of a graph D is said to be *intra-dataset* if $\lambda(source(l)) \in \Delta_D, \lambda(target(l)) \in \Delta_D$, otherwise it is said to be *inter-dataset*.

Within a dataset graph, edges connecting two external nodes are simply ignored to avoid possibility of link spam. Any dataset could possibly create an arbitrary number of links between two external entities which may lead to anomalies in the graph and affect the link analysis algorithms.

2.2 PageRank

PageRank [3] is a ranking system that originates from works on Web search engines. The ranking system, based on a random walk algorithm, evaluates the probability of finding a random web surfer on any given page. The algorithm assumes a hyperlink from a page i to a page j as an evidence of the importance of page j. In addition, the amount of importance that i is conferring to j is determined by the importance of i itself and inversely proportional to the number of pages i points to. PageRank can easily be adapted to the Web of Data model. By regarding pages as entities and hyperlinks as links between entities, we can formulate PageRank as follow:

Let $L(i) = \{target(l) | \forall l \in L, source(l) = i\}$ be the set of entities linked by an entity i and $B(j) = \{source(l) | \forall l \in L, target(l) = j\}$ be the set of entities that points to j. The PageRank $r(j)$ of an entity $j \in E$ is given by:

$$r^k(j) = \alpha \sum_{i \in B(j)} \frac{r^{k-1}(i)}{|L(i)|} + \frac{(1-\alpha)}{|E|} \ . \tag{1}$$

A fixed-point iteration approach is commonly used where the computation of a rank score $r^k(j)$ at a step k uses the rank scores of the step $k-1$. The operation is repeated until all scores r stabilise to within some threshold.

The PageRank formula is composed of two parts weighted by a damping factor α, usually set to 0.85. The first component provides the contribution $\sum_{i \in B_j} \frac{r^{k-1}(i)}{|L(i)|}$ of incoming links to entity j. The factor $\frac{1}{|L(i)|}$ defines a uniform distribution of the importance of entity i to all the entities i points to. This distribution, later referred to as $w_{i,j}$, can be modified in order to provide a distribution based on the weight of a link. The second component $\frac{1}{|E|}$ denotes the probability that the surfer will jump to j from any other random entity from the collection.

3 Related Work

Link analysis such as PageRank [3] has been successfully applied for query independent ranking (also called static ranking). Several extensions have been developed to take into consideration weighted links, hierarchical link structure or the Semantic Web model.

Weighted Link Analysis. When working with more heterogeneous links, standard approaches do not provide accurate results since links of different types can have various impact on the ranking computation. In [5, 6], the authors extend PageRank to consider different types of relations between entities. PopRank [7], an object-level link analysis, proposes a machine learning approach to assign a "popularity propagation factor" to each type of relation. ObjectRank [8] applies authority-based ranking to keyword search in databases. However, these works do not consider the features of links such as their specificity and cardinality to assign weights in an unsupervised fashion. Furthermore, these approaches are too complex to apply on web-scale since they will require multiple times the current processing power of current web search engines. A major task is to bring down this computational power requirement.

Hierarchical Link Analysis. Recent works [9–14] exploit the hierarchical structure of distributed environments and of the Web. [9] suggests a hierarchical model of the Web and shows the desirable computational properties of such approach. In [12], the authors show that hierarchical ranking algorithm outperforms qualitatively other well-known algorithms, including PageRank. However, such models have never been applied on semi-structured data sources with distinct semantics and none of them are considering weighted relations between supernodes.

Semantic Web Link Analysis. SemRank [15] proposes a method to rank semantic relations using information-theory techniques but is solely focussed on ranking and retrieval of relationships. The Swoogle search engine [1] is the first one to propose OntoRank, an adaptation of PageRank for Semantic Web resources. In ReconRank [2], a link analysis is applied at query time for computing the popularity of resources and documents. The above algorithms only consider the individual web resources, disregarding the semantics and structure of the Web of Data. They are therefore costly on a web-scale and likely provide sub-optimal results. We are aware of one recent study [4] that has analysed the effectiveness of PageRank on the domain level for ranking Semantic Web resources. However, their approach disregards the link structure between entities within a domain, and does not consider weighted relations.

Our Contribution. To address the Web of Data scenario previously described, we first illustrate a two-layer model of the Web of Data. We introduce and evaluate a hybrid algorithm that combines both weighted and hierarchical link analysis methods. The model operates in a hierarchical fashion between a dataset and entity layer leveraging an unsupervised method that considers both the specificity and cardinality of links for assigning them appropriate weights. First an extension of weighted PageRank is applied on the dataset layer. Then, the importance of each dataset node is distributed to its individuals entities, and combined with local entity ranks which can be dependent of the dataset semantic.

4 A Two-Layer Model for Ranking Web Data

In this section, we introduce a two layer model for the Web of Data, pictured in Fig. 1. The top layer (dataset layer) is composed of a collection of inter-connected datasets whereas the lower layer (entity layer) is composed of independent graphs of entities.

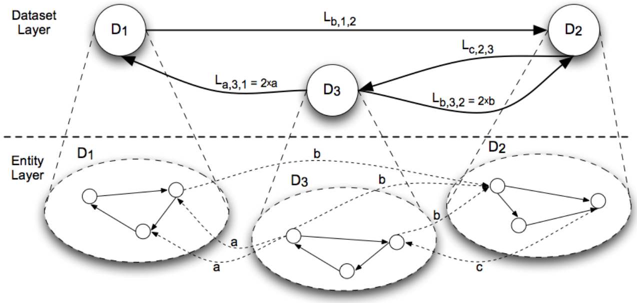


Fig. 1. The two-layer model of the Web of Data. Dashed edges on the entity layer represent inter-dataset links.

4.1 Quantifying the Two-Layer on the Web of Data

In this section, we provide evidence of the two-layer model and its desirable computational properties by quantifying the locality characteristics of links and the dataset size distribution. We perform the following simple experiments. We first take the datasets described below and count how many of the links are intra-dataset and how many are inter-dataset. Then, we analyse the dataset size distribution on a subset of the Web of Data.

DBpedia is a semi-structured version of Wikipedia and contains 17.7 million of entities[5].

Citeseer is a semi-structured version of Citeseer from the RKBExplorer initiative and contains 2.48 million of entities[6] .

[5] DBpedia: `http://dbpedia.org/`

[6] Citeseer: `http://citeseer.rkbexplorer.com/`

Geonames is a geographical database and contains 13.8 million of entities[7].

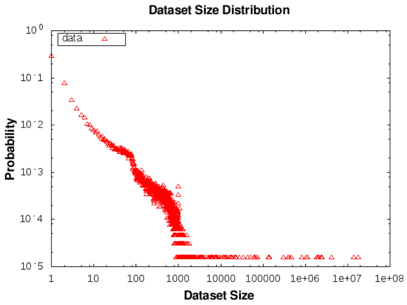
Sindice's Page-Repository contains 60 million of entities among 50.000 datasets (including the previous). It is a representative subset of the Web of Data. It is composed of Semantic Web online repositories and pages with microformats or RDFa markups crawled on a regular basis for more than two years[8].

Table 2(a) shows that 78.8% of the links are intra-dataset. Such connectivity statistics are not far from the previous results of [9] where 83.9% of the links from a large web pages dataset are intra-domain links. On individual datasets, inter-dataset links in DBpedia represent only 6.8% of its total links. For Citeseer, the number of inter-dataset links is higher than other datasets but can be explained by the fact that this dataset is using an external ontology to describe its data, hence most of its inter-dataset links point to only one external dataset (the dataset ontology). Geonames is representative of a "dataset sink", a dataset loosely linked with other datasets. These numbers confirm a high degree of locality on the Web of Data, and suggest the two-layer model proposed in this paper.

Fig. 2(b) depicts the distribution of the size of all datasets found in Sindice's page-repository. The distribution nearly follows a powerlaw and corresponds to previous research on the size of web sites [13]. We observe that the majority of the datasets contain less than 1000 nodes which indicates that local rank computation within these graphs can be performed in an efficient manner in memory.

Dataset	***Intra***	***Inter***
Full	287M (78.8%)	77M (21.2%)
DBpedia	88M (93.2%)	6.4M (6.8%)
Citeseer	12.9M (77.7%)	3.7M (22.3%)
Geonames	59M (98.3%)	1M (1.7%)

(a) Ratio intra / inter dataset links



(b) Distribution of the size of datasets

Fig. 2. Statistics about link locality and dataset size

4.2 The Dataset Graph

The top layer, or dataset graph, can be seen as an approximation of the global graph G. Instead of considering entities and links between these entities, we are using higher-level information such as datasets and *linksets*. Given a dataset D, we denote a linkset with $L_{\sigma,i,j} = \{l | label(l) = \sigma, source(l) \in D_i, target(l) \in D_j\}$

[7] Geonames: `http://www.geonames.org/`

[8] sindice: `http://www.sindice.com/`

the set of edges having the same label σ and connecting the dataset D_i to the dataset D_j. For example, in Fig. 1 the inter-dataset links (dashed-edges labelled a) between D_3 and D_1 are aggregated to form the linkset $L_{a,3,1}$ on the dataset layer.

The resulting graph (50.000 nodes, 1.2M of linksets) is orders of magnitude smaller than the original graph G (60M nodes, 364M of links). As a consequence, it can be easily kept in memory (in the case of Sindice) and dataset ranks can be computed on demand.

4.3 The Entity Graph

The lower layer, or entity graph, is composed of disjoint graphs D each of them being a collection of internal nodes and intra-dataset edges. The direct consequence is that the computation of ranks at entity level can be computed in an independent manner (on a per dataset basis) and can be easily parallelised. Since computations are performed independently, the complexity that would dominate is that of the largest dataset, e.g. DBpedia in our case. However, the majority of the graphs has a small number of nodes as shown in Fig. 2(b). This means that the graphs can be easily kept in memory and rank computation can be performed without the performance penalty of IO access generally encountered when processing very large graphs.

5 The DING Model

In this section, we start by introducing an unsupervised method for weighting links and linksets. Next, the DING algorithm is described. We first reformulate the original PageRank algorithm for computing dataset ranks. We present two local entity ranking algorithms as well as a list of semantic-dependent algorithms that is known to outperform standard algorithms for certain type of dataset. We finally explain how to combine dataset ranks with local entity ranks in order to estimate a global entity ranking.

5.1 Unsupervised Link Weighting

In Fig. 1 the probability of the user going from D_3 to D_1 is likely to be different from the probability of going to D_2 since the label and number of links associated to $L_{a,3,1}$ are not the same as the ones associated to $L_{b,3,2}$. The goal is to define a linkset weighting function $w_{\sigma,i,j}$.

Weights can be assigned based both on the number of links contained in a linkset and on the general importance of the label involved in the link. Our approach is derived from TF-IDF to measure the relevance of a label given its frequency in a data collection. We define the linkset weighting function w using the Link Frequency - Inverse Dataset Frequency (LF-IDF):

$$w\sigma, i, j = LF(L_{\sigma,i,j}) \times IDF(\sigma) = \frac{|L_{\sigma,i,j}|}{\sum_{L\tau,i,k} |L_{\tau,i,k}|} \times \log \frac{N}{1 + freq(\sigma)} \ . \tag{2}$$

In Eq. (2), N denotes the number of datasets and $freq(\sigma)$ is the frequency of occurrence of the label σ in the collection of datasets. These weights can be computed dynamically given accumulated statistical information in the database. The LF-IDF scheme assigns a higher degree of importance to link with a high frequency (in a given dataset) and low dataset frequency in the dataset collection. For example, this weighting scheme will favour links such as `foaf:knows` compared to very frequent links such as `rdfs:seeAlso`. Former results [16] have shown that link weighting improves dataset ranking.

5.2 DING Algorithm

The DING algorithm is an extension of PageRank (Eq. 1) for the two-layer graph structure presented in Sect. 2.1. Instead of visiting web pages, the random surfer browses datasets. The random walk model is as follows:

1. At the beginning of each browsing session, a user randomly selects a dataset.
2. Then, the user may choose one of the following actions:
   (a) Selecting randomly an entity in the current dataset.
   (b) Jumping to another datasets that is linked by the current dataset.
   (c) Ending the browsing.

According to the hierarchical random walk model, we can apply a two-stage computation. In the first stage, we calculate the importance of the top level dataset nodes which is explained next. The second stage calculates the importance of entities within a dataset, as explained in Sect. 5.4.

5.3 Computing DatasetRank

The dataset surfing behaviour is the same as in PageRank. We can obtain the importance of dataset nodes by applying PageRank on the weighted dataset graph.

As in (1), the rank score $r(D_j)$ of a dataset is composed of a part corresponding to the rank contribution from the datasets linking to D_j and of a part corresponding to the probability of a random jump to D_j from any dataset in the collection. The probability of selecting a dataset during a random jump is proportional to its size, i.e. $|E_{D_j}|$. The distribution factor $w_{\sigma,i,j}$ is defined by Eq. 2. The final DatasetRank formula is given below. The two parts are combined using the damping factor $\alpha = 0.85$, since we observed that this value provides also good results in our experimental evaluation.

$$r^k(D_j) = \alpha \sum_{L\sigma,i,j} r^{k-1}(D_i) w_{\sigma,i,j} + (1-\alpha)\frac{|E_{D_j}|}{\sum_{D\in\mathcal{G}} |E_D|} \ . \tag{3}$$

5.4 Computing Local Entity Rank

A method used in layered ranking algorithms is to assign to the page the importance of the supernode [10, 4]. In our case this would correspond to assign the DatasetRank score of a dataset to all its entities. In large datasets, such as DBpedia, this approach does not hold. A query is likely to return many entities from a same dataset with the same rank. This unnecessarily pushes part of the ranking problem at query time. Instead we can assign a score combining both the importance of a dataset and the importance of an entity within its dataset.

Next, we present two generic algorithms, the weighted EntityRank and the weighted LinkCount, that compute entity ranks on any type of graphs. However, we argue that entity ranking is strongly dependent of the semantic of the dataset. A list of existing semantic-dependent algorithms is discussed afterwards.

Weighted EntityRank The Weighted EntityRank method uses the PageRank algorithm from Eq. 1 applied on the internal entities and intra-links of a dataset in order to compute the importance of an entity node within a dataset. In our experimental setup, we use the LF-IDF weighting scheme from Eq. 2 on single links between entities. Like PageRank, the robustness against spam of the EntityRank method makes it a good choice for datasets build on non-controlled user inputs like Livejournal or Last.fm.

Weighted LinkCount The Weighted LinkCount is a variant of the *in-degree counting links* method [17], an alternative to EntityRank when the dataset can be assumed mostly deduplicated and spam-free. This is often true for very well curated user-input datasets like DBpedia. For each entity j, its rank $r(j)$ is given by $r(j) = \sum_{l_{\sigma,i,j}} w(l_{\sigma,i,j})$ where $w(l_{\sigma,i,j})$ is the weight of the link from i to j. The weighting scheme is the same as the one used in EntityRank. LinkCount is more efficient to compute than EntityRank, since it needs only one “iteration” over the data collection.

Semantic-Dependent Entity Ranking Datasets on the Web of Data may have their own semantic with a variety of graph structures. For example, we can mention generic graphs coming from user inputs, hierarchical graphs, bipartite graphs, etc. A complete taxonomy of the different graph structures among existing datasets is beyond the scope of the paper, but several examples are presented in Tab. 1.

While EntityRank and LinkCount represent good generic solutions for local entity ranking, as shown in Sect. 7.2, an approach which takes into account the peculiar properties of each dataset will give better results. Considering that in literature there are already a notable amount of ranking algorithms that are dataset specific, such as [18, 19] for citation networks or Dissipative Heat Conductance [12] for strongly hierarchical datasets like taxonomies or geo-databases, DING has been designed to exploit better alternatives to LinkCount and EntityRank. One can also define its own algorithm using dataset-dependent ranking criteria. If the given local entity ranking is modelled as a probability distribution, combining it with DatasetRank means simply calculating the joint probability as explained next.

Graph Structure	***Dataset***	***Algorithm***
Generic, Controlled	DBpedia	LinkCount
Generic, Open	Social Communities	EntityRank
Hierarchical	Geonames, Taxonomies	DHC
Bipartite	DBLP	CiteRank

Table 1. List of various graph structures with targeted algorithms

5.5 Combining DatasetRank and Entity Rank

A straightforward approach for combining dataset and local entity ranks is to adopt a purely probabilistic point of view by interpreting the dataset rank $r(D)$ as the probability of selecting the dataset and the local entity rank $r(e)$ as the probability of selecting an entity within this dataset. Hence we would have the global score $r_g(e)$ defined as $r_g(e) = P(e \cap D) = r(e) * r(D)$.

But this approach favours smaller datasets. The local entity ranks is much higher in small datasets than in larger ones, since in the probabilistic model all ranks in a dataset sum to 1. As a consequence any small dataset receiving even a single link is likely to have its top entities score way above many of the top ones from larger datasets. The solution is to normalize the local ranks to a same *average* based on the dataset size. In our experiments we use the following formula for ranking an entity e in a dataset D: $r_g(e) = r(D) * r(e) * \frac{|E_D|}{\sum_{D' \in G} |E'_D|}$.

6 Scalability of the DING approach

A precise evaluation of the scalability of our approach is not the goal of this paper. Moreover, [9] has shown that hierarchical ranking algorithms provide speedup in computation compared to standard approaches. However, we report here some performance results from the DING method when applied on a real use-case scenario, i.e., the Sindice search engine.

Given the small size of the dataset graph as shown in Sect. 4.2, the graph can be fully held in memory and rank computation can be performed on demand. A single iteration of DatasetRank computation takes 200ms to process 50k datasets on commodity hardware (Intel Xeon E5410 Quad Cores), a good quality rank can hence be obtained in a matter of seconds. If we define a measure of convergence of the algorithm at an iteration $k+1$ as in Eq. 4, the algorithm converges to a 0.1% threshold in 32 iterations, which represents 5 seconds.

$$\rho(k+1) = \max_{D_i \in \mathcal{D}} \frac{|r^{k+1}(D_i) - r^k(D_i)|}{r^k(D_i)} \tag{4}$$

Since the size of the majority of the datasets is in order of thousands of nodes as shown in Sect. 4.3, their entity graph can also be held entirely in memory making more effective the computation of entity ranks. Moreover, since the computation of entity ranks in one dataset is independent of the entity rank from another datasets, the computation can be easily distributed over a cluster of machines.

On the other hand, the computation of entity ranks in large datasets can become a heavy operation considering that the largest dataset (i.e., DBpedia) is containing over ten million entities and tenths of millions links. For such dataset, we fall back on standard methods to parallelise the computation such as the *Map-Reduce* programming model [20]. Computing[9] the local entity ranks of the DBpedia dataset with a 0.1% precision took 55 iterations of 1 minute each on a Map-Reduce cluster composed of three Intel Xeon quad cores. In the case of LinkCount, the computation would require only one such iteration.

In addition, separating dataset ranks and local entity ranks minimizes the amount of computation required when updating the data collection. For example, a new dataset D_i which has links to several other datasets has to be indexed by Sindice. With standard non-hierarchical ranking models, the ranks of all entities would have to be recomputed. In contrast, with the DING model the only set of ranks to be computed are 1. the ranks of entities in dataset D_i; and 2. the dataset ranks which can be recomputed in a matter of seconds. This decreases the cost of the update from being proportional to the size of the Web of Data to being proportional to the size of the dataset D_i and of the dataset graph.

7 Experiments and Results

We introduced a novel ranking model, showed that it can cope with dataset semantics and gave evidences about its desirable computational properties. But it is not yet clear if the DING model provides worst, similar or better performance than standard approaches. In order to assess the performance of DING, we conduct two experiments. The baseline algorithm that we use for comparison is a global version of EntityRank (GER). This algorithm is similar to the one described in Sect. 5.4 with the only difference that it operates on the full Web of Data graph G. We use the datasets presented in Sect. 4.1 for the two experiments.

The first experiment investigates the impact of link locality on the Web of Data by comparing the performance of the two generic local algorithms, the local EntityRank (LER) and local LinkCount (LLC), with GER. This first experiment is done without user intervention by measuring the rank correlation between the algorithms. While this experiment provides strong evidence of the effectiveness of LLC and LER, it does not assess the quality of retrieval. The second experiment evaluates the effectiveness of the local algorithms and of DING through a user study in order to judge if they provide worst, similar or better performance than the baseline approach.

7.1 Accuracy of Local Entity Rank

The goal of this experiment is to compare the static ranks in a query independent manner of the local algorithms (LER and LLC) with the global one (GER) in order to judge the impact of the link locality on the Web of Data. We measure

[9] including Input/Output disk operations as well as data preprocessing

the Spearman's correlation [21] of the two local algorithms with GER using the full entity rank list of three datasets: DBpedia, Citesser and Geonames. The Spearman's correlation has already been employed in [18] to compare several ranking algorithms. The results are presented in Table 2[10]. While LLC performs slightly worse than LER, the Spearman's correlation indicates a strong correlation of the two algorithms with GER. These results confirm that, on individual datasets, GER can be well approximated with computational methods of lower complexity such as LER or LLC due to the high degree of link locality.

Algorithm	***DBpedia***	***Citeseer***	***Geonames***
LLC	0.79	0.86	0.73
LER	0.88	0.97	0.78

Table 2. Spearman's correlation between LLC and LER with GER

7.2 User Study

Information Retrieval experiments focus on retrieval effectiveness, expressed in terms of recall and precision. In general, data collections such as the one provided by TREC or INEX are employed to assess the ranking produced by systems. The TREC or INEX entity tracks are corpus created for the evaluation of entity-related searches. However, they are not suitable for our use cases where queries of various complexity are used and where the goal is to measure the effectiveness of ranking among inter-linked datasets. Therefore, in order to evaluate qualitatively the DING methodology, we decided to perform a user study where users provide relevance judgements for each algorithm.

Design The user study is divided into two experiments: 1. the first one (Exp-A) assesses the performance of local entity ranking on the DBpedia dataset; 2. the second one (Exp-B) assesses the performance of local entity ranking on the full Sindice's page-repository. Each experiment includes 10 queries, varying from simple keyword search to more complex structured queries (SPARQL). Each participant receives a questionnaire containing a description of the query in human language, and three lists of top-10 results. Each result is described by the human-readable label and the URI of the entity. The first list corresponds to the ranking results of GER. For Exp-A, the second and third lists correspond to the ranking results of LER and LLC while for Exp-B the ranking results are from DatasetRank combined with LER and LLC (DR-LER and DR-LLC resp.). The second and third lists are named randomly "Ranking A" or "Ranking B" so no information about the ranking algorithm and no correlation between Ranking A and B on two questionnaires can be inferred. For these experiments, we consider the effect of link-based features in combination with textual features. Therefore, the three lists of results are ordered using both the static rank from the link

[10] The Spearman's correlation coefficient tests the strength of the relationship between two variables, i.e. ranks produced by LER or LLC and GER. The values varies between 1 (a perfect positive correlation) and -1 (a perfect negative correlation). A value of 0 means no particular correlation.

analysis algorithms and a query-dependent ranking (similar to BM25), combined using a simple linear combination.

Participants Exp-A evaluation is performed on 31 participants, and Exp-B evaluation on 58 participants. The participants consist of researchers, doctoral and master students and technicians. All of the participants are familiar with search engines, but a few of them familiar with entity search engines.

Task The task is to rate "Ranking A" in relation to the standard one using categorical variable, then to rate "Ranking B" in relation to the standard one. The participants have to choose between 5 categories: Better (B), Slightly Better (SB), Similar (S), Slightly Worse (SW), Worse (W). The questionnaires and the raw results of the user study can be downloaded at `http://ding.sindice.com/`.

Measure We use the Pearson's chi-square to perform the test of "goodness of fit" between O, the observed frequency distribution (the participant's judgements) of the previous categories, and E, an expected theoretical uniform distribution (equiprobable) of these categories, in order to establish whether or not the observed distribution differs from the theoretical distribution. Our null hypothesis is that the observed frequency distribution is uniform. We then interpret the contribution to chi-square of each category.

Exp-A Results The tables 3(a) and 3(b) show the results[11] of the chi-square test for LER and LLC respectively. For the tests to be significant at the 1% level, with 4 degrees of freedom, the value for chi-square has to be at least 13.3. Since the chi-square test yields 49.48 for LER and 14 for LLC, we can reject the null hypothesis for the two tests. It bears out that a large proportion of the population (+71% of contribution to χ^2) considers LER similar to the GER. For LLC, a majority of the population (+53% of contribution to χ^2) considers it similar to GER, and this is reinforced by the fact that a minority (−31% of contribution to χ^2) considers it worse.

To conclude, at 1% significance level, LER and LLC provides similar results than GER. However, there is a more significant proportion of the population that considers LER more similar to GER.

Exp-B Results The tables 3(c) and 3(d) show the results of the chi-square test for DR-LER and DR-LLC respectively. For the tests to be significant at the 1% level, with 4 degrees of freedom, the value for chi-square has to be at least 13.3. Since the chi-square test yields 16.31 for DR-LER and 20.45 for DR-LLC, we can reject the null hypothesis for the two tests. It bears out that a good proportion of the population (+57% of contribution to χ^2) considers DR-LER similar to GER, strengthen by the fact that a minority (−39% of contribution to χ^2) considers it worse. For DR-LLC, a large proportion of the population (+65%

[11] Intermediate calculation steps are omitted.

(a) LER

Rate	O_i	E_i	$\%\chi^2$
B	0	6.2	−13%
SB	7	6.2	+0%
S	21	6.2	+71%
SW	3	6.2	−3%
W	0	6.2	−13%
Totals	31	31	

(b) LLC

Rate	O_i	E_i	$\%\chi^2$
B	3	6.2	−12%
SB	8	6.2	+4%
S	13	6.2	+53%
SW	6	6.2	−0%
W	1	6.2	−31%
Totals	31	31	

(c) DR-LER

Rate	O_i	E_i	$\%\chi^2$
B	12	11.6	+0%
SB	12	11.6	+0%
S	22	11.6	+57%
SW	9	11.6	−4%
W	3	11.6	−39%
Totals	58	58	

(d) DR-LLC

Rate	O_i	E_i	$\%\chi^2$
B	7	11.6	−9%
SB	24	11.6	+65%
S	13	11.6	+1%
SW	10	11.6	−1%
W	4	11.6	−24%
Totals	58	58	

Table 3. Chi-square test for Exp-A and Exp-B. The column $\%\chi^2$ gives, for each modality, its contribution to χ^2 (in relative value).

of contribution to χ^2) considers it slightly better than GER, which is comforted by the fact that a minority (−24% of contribution to χ^2) considers it worse.

To conclude, at 1% significance level, the two algorithms give a profile of preference quite different. It appears that DR-LLC provides a better effectiveness. Indeed, a large proportion of the population finds its results slightly better than GER, and this is reinforced by a few number of people finding it worse.

8 Conclusion and Future Work

We presented DING, a novel two-layer ranking model for the Web of Data. DING is specifically designed to address the Web of Data scenario, computing the popularity score of entities on web-scale graph. As opposed to alternative approaches, we explain its desirable computational properties and display experimental evidence of improved ranking quality. Furthermore, since DING allows for improved local ranking by using dataset-specific ranking algorithms, further works will be done in the area of automation of graph structure recognition. This would allow better match of specific ranking algorithms to a graph semantic, and will improve the performance of the ranking on a heterogeneous Web of Data.

9 Acknowledgments

This material is based upon works supported by the European FP7 project *Okkam - Enabling a Web of Entities* (contract no. ICT-215032), and by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Ding, L., Pan, R., Finin, T.W., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. In: Proceedings of the International Semantic Web Conference. (2005) 156–170
2. Hogan, A., Harth, A., Decker, S.: Reconrank: A scalable ranking method for semantic web data with context. In: Proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems, Athens, GA, USA. (11 2006)
3. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999)

4. Harth, A., Kinsella, S., Decker, S.: Using Naming Authority to Rank Data and Ontologies for Web Search . In: The Semantic Web - ISWC 2009. Volume 5823 of Lecture Notes in Computer Science., Berlin, Heidelberg, Springer Berlin Heidelberg (2009) 277 – 292
5. Xing, W., Ghorbani, A.: Weighted pagerank algorithm. In: CNSR '04: Proceedings of the Second Annual Conference on Communication Networks and Services Research. Volume 0., Washington, DC, USA, IEEE Computer Society (2004) 305–314
6. Baeza-Yates, R., Davis, E.: Web page ranking using link attributes. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM (2004) 328–329
7. Nie, Z., Zhang, Y., Wen, J.R., Ma, W.Y.: Object-level ranking: bringing order to Web objects. In: Proceedings of the 14th international conference on World Wide Web, ACM (2005) 567
8. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: authority-based keyword search in databases. In: Proceedings of the Thirtieth international conference on Very large data bases, VLDB Endowment (2004) 564–575
9. Kamvar, S., Haveliwala, T., Manning, C., Golub, G.: Exploiting the block structure of the web for computing pagerank. Technical Report 2003-17, Stanford InfoLab (2003)
10. Eiron, N., McCurley, K.S., Tomlin, J.A.: Ranking the Web Frontier. In: Proceedings of the 13th conference on World Wide Web. Number 2, New York, New York, USA, ACM Press (2004) 309–318
11. Wang, Y., DeWitt, D.J.: Computing pagerank in a distributed internet search system. In: Proceedings of the Thirtieth international conference on Very large data bases, Toronto, Canada, VLDB Endowment (2004) 420–431
12. Xue, G.R., Yang, Q., Zeng, H.J., Yu, Y., Chen, Z.: Exploiting the hierarchical structure for link analysis. In: Proceedings of the 28th annual international ACM SIGIR conference, New York, NY, USA, ACM (2005) 186–193
13. Feng, G., Liu, T.Y., Wang, Y., Bao, Y., Ma, Z., Zhang, X.D., Ma, W.Y.: Aggregaterank: bringing order to web sites. In: Proceedings of the 29th annual international ACM SIGIR conference, ACM Press (2006) 75
14. Broder, A.Z., Lempel, R., Maghoul, F., Pedersen, J.: Efficient pagerank approximation via graph aggregation. Information Retrieval **9** (2006) 123–138
15. Anyanwu, K., Maduko, A., Sheth, A.: Semrank: ranking complex relationship search results on the semantic web. In: Proceedings of the 14th international conference on World Wide Web, ACM (2005) 117–127
16. Toupikov, N., Umbrich, J., Delbru, R., Hausenblas, M., Tummarello, G.: DING! Dataset Ranking using Formal Descriptions. In: WWW 2009 Workshop: Linked Data on the Web (LDOW2009), Madrid, Spain (2009)
17. Najork, M.A., Zaragoza, H., Taylor, M.J.: Hits on the web: how does it compare? In: Proceedings of the 30th annual international Annual ACM Conference on Research and Development in Information Retrieval. (2007)
18. Sayyadi, H., Getoor, L.: Futurerank: Ranking scientific articles by predicting their future pagerank. In: SDM. (2009) 533–544
19. Walker, D., Xie, H., Yan, K.K., Maslov, S.: Ranking scientific publications using a simple model of network traffic. CoRR (2006)
20. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM **51**(1) (2008) 6
21. Melucci, M.: On rank correlation in information retrieval evaluation. SIGIR Forum **41**(1) (2007) 18–33