SPRING:
Ranking the results of SPARQL queries on Linked Data

Kunal Mulay and P. Sreenivasa Kumar

Indian Institute of Technology, Madras
Chennai, India
{kunalm,psk}@cse.iitm.ac.in

Abstract

Linked Open Data (LOD) is a huge effort in the direction of making the Web of Data a reality. The LOD cloud consists of about 200 datasets contributed by several independent data providers from various domains such as Publications, Geography, Government, Media, Biology and Drugs etc. The data is represented in RDF and SPARQL is the query language. Due to the open nature of the data and its heterogeneous origin, it is often the case that a real-world entity appears in different datasets and with different names. When such entities are to be reported in the query results, a mechanism of rank ordering them becomes essential. In this paper, we propose a new framework for calculating the importance scores of datasets and also resources inside the datasets. As consensus is a key element that adds value to the data in the Web of Data, we base our ranking framework on constructs that are used to express *sameness* or *equivalence* among the entities in RDF data. We also make use of the underlying graph structure of LOD in the framework. The framework is experimentally verified on the Billion Triple Challenge (BTC-2010) dataset. The results indicate that the framework is successful in giving entities from the most relevant dataset a higher score compared to other datasets.

1 Introduction

The web of data[2] has attracted a lot of attention from researchers in the recent past. Web of data represents individual entities and properties by their own URIs. Unlike in the case of web of documents, the links in web of data are "typed", as links are labeled by property names. The Linked Open Data(LOD)[1] project represents a huge effort in the direction of realizing the web of data. The data in the LOD cloud has grown tremendously in last few years and it currently consists of 256 number of datasets and 30,874,103,042 number of triples(or links)[2].

When compared to traditional web, the web of data contains datasets created by various research organizations, companies and individuals. It is often the case that an entity appears in several datasets with different names(URIs). The entity class names and the property names used in a dataset constitute a shared vocabulary and enable the dataset publishers to convey semantic information about the data. The current major search engines, namely Google, Yahoo and Bing, recognize the need of adopting standard vocabularies for annotating text data. They are promoting the adoption of shared vocabularies for commonly used entities through `www.schema.org`. There are also other efforts such as *Microformats*[3] and *RDFa*[4] in this direction.

The web of data community uses Resource Description Framework(RDF)[5] as a standard for data representation. XML[6] is used as one of the mechanisms for representing RDF data. In addition, Notation3 (N3)[7], N-Triples[8] and Turtle[9] serialization formats also exist for representing RDF data. In RDF, each data item is represented by a triple (subject - predicate - object). Here subject and object represent individual resources(entities) and predicate represents a relation(link) between them. When represented in the form of a graph, subject and object become the nodes and predicate becomes a directed edge from subject node to object node. Here, each of the resources (subject, predicate and object) are named with URI references, with the exception that the value of an object can either be a URI reference or a string literal. For example, the RDF representation about the phone

17^{th} *International Conference on Management of Data*
COMAD 2011, Bangalore, India, December 19–21, 2011

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

[2] `http://www4.wiwiss.fu-berlin.de/lodcloud/state/`
[3] http://microformats.org
[4] http://www.w3.org/TR/xhtml-rdfa-primer
[5] http://www.w3.org/TR/rdf-concepts
[6] http://www.w3.org/TR/REC-rdf-syntax
[7] `http://www.w3.org/DesignIssues/Notation3.html`
[8] `http://www.w3.org/TR/rdf-testcases/#ntriples`
[9] `http://www.w3.org/TeamSubmission/turtle/`

"HTC HD7S" could include the following triples:
(product1, hasName , "HTC HD7S")
(product1, hasManufacturer, HTC)
(product1, hasNetwork, 2G)
(product1, hasNetwork, 3G)
(Product1, hasOS , Windows7Mobile)
(product1, isSuccessorOf, product2)
(product2, hasName, "HTC HD7")
Here, first and last triples use string literal as object values. All the other names used are in fact URIs(they are omitted here for brevity).

SPARQL[10] is a W3C standard for querying RDF repositories. Considering RDF repository as a RDF graph, a SPARQL query returns the nodes that satisfy certain conditions. These conditions are formed using triple patterns, and the collection of triple patterns is called Basic Graph Pattern (BGP). For example, below is the SPARQL query to find name of a mobile which is manufactured by HTC and supports 3G network.
Select ?name Where {
?product <hasManufacturer> <HTC> .
?product <hasNetwork> <3G> .
?product <hasName> ?name }
Here each row inside the *where* clause shows a triple pattern and the collection of all triple patterns inside the *where* clause represents a basic graph pattern. When this SPARQL query is run against the above set of triples, the variable *product* is bound with <product1> node and the variable *name* is bound with "HTC HD7S".

Due to the open nature of web of data and also its size, a typical SPARQL query results in several answers. For example if we query for a researcher affiliated with a particular institute and working on a particular topic, we might get results from DBLP dataset[11], Freebase[12], DBpedia[13] and also some other datasets. In such a scenario, it becomes necessary to rank order the results of SPARQL queries. In this paper, we focus on this problem. Consensus or agreement plays a vital role in adding value to data in the context of web of data. We seek to utilize the features of web of data that promote consensus to propose a new framework of ranking for resources and triples. We present a complete system called **SPRING** (**SP**ARQL **R**esult rank**ING**). To the best of our knowledge, no one has investigated ranking of SPARQL results.

The main contribution of this paper are: A study of ranking frameworks for web of data, a domain independent, three level architecture and a global ranking system for all datasets in Linked Data cloud.

[10]http://www.w3.org/TR/rdf-sparql-query
[11]`http://dblp.l3s.de/d2r/`
[12]`http://www.freebase.com`
[13]`http://dbpedia.org`

2 Related work

Link Analysis. In traditional web setting, link analysis is used by various search engines to rank relevant pages. Link analysis is of interest for the researchers working in web IR, social networks and semantic web. In web, two pages are connected by a hyperlink. While in social network and semantic web, two entities are connected by a link which represents a relationship. Ranking algorithms like PageRank[13] and HITS[11] make use of existing link structure for ranking.

HITS is an authority based algorithm, which assigns two scores to each page. The authority score calculates the value of content on that page, while the hub score calculates the value of links to the other pages. The algorithm initializes these values to 1 and authority and hub update is performed iteratively until the value converges.

The PageRank algorithm is built upon random surfer model. In this model a surfer begins with browsing a random URL. After visiting the page, the surfer can either navigate to one of the link on the page with probability d or jump to a random URL with probability $1 - d$. The PageRank algorithm assigns a constant value to all the pages and iteratively propagates the values until convergence.

Ranking in Semantic Web. Ranking in semantic web is limited to ontology ranking, relationship ranking and entity ranking.
Swoogle[4] proposed an ontology rank algorithm for indexing and searching ontologies or Semantic Web Documents(SWD). Here N-gram is used for matching SWDs. It uses a rational surfer model which has roots in random surfer model. Later they extend the search to Semantic Web Terms(SWT) or entities[5].

PopRank[12] is an object level ranking algorithm, which extends PageRank model by adding popularity propagation factor to the links pointing towards the objects. It uses a supervised learning to rank objects. The training data is prepared by domain experts and thus the algorithm is limited to a particular domain.

SemRank[1] proposes a relevance model, which uses semantic and information-theory techniques. But it is only focused on the relationships and not other resources.

ReConRank[8] is divided into two parts, ResourceRank and ContextRank. Here each of the algorithm uses PageRank model for their calculation. ResourceRank is applied over the resource graph and ContextRank is applied over the context graph. Later these two are combined to form ReConRank. The model uses intelligent surfer model as opposed to other approaches which uses random surfer model.

TripleRank[6] is built on top of HITS algorithm. It converts the semantic web graph into a three-dimensional tensor representation. Then the authority ranking algorithm is applied to rank the resources.

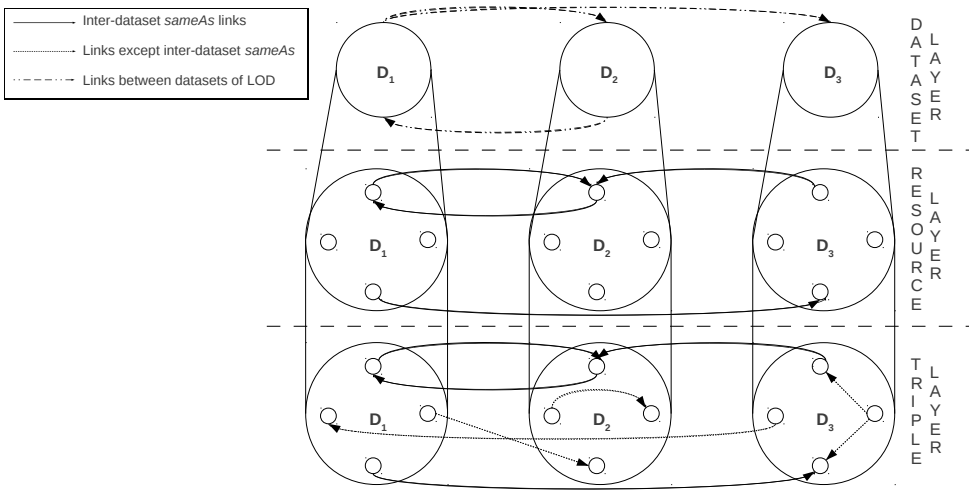

Figure 1: A three-layer model for ranking SPARQL query results

The limitations of this model is maintaining the tensor when applied to web of data.

DING [3] uses a two-layered random surfer model for ranking RDF resources. Here one layer is for ranking datasets and other for entities. It modified the existing TF-IDF algorithm and proposed LF-IDF (Link Frequency - Inverse Dataset Frequency) algorithm to measure relevance of a link label given its frequency in the data.

A recent paper discusses ranking Linked Data from relational database[10]. The algorithm focused only on Linked Data generated by relational databases. It converts the data to RDF using D2R server, then calculates rank score of the relationships using concept-hops. The algorithm fails to perform when the dataset contains explicit relationships. In which case, the number of concept-hops counts to minimum and all relationships have same score.

In addition there are some more ranking algorithms for ranking ontologies and resources but they revolve around the above discussed models. To our knowledge no model is designed to rank SPARQL query results.

At this point it is relevant to mention the differences between web query result ranking and ranking of SPARQL query results. Currently, the leading search engines use a variety of parameters for ranking the web query results. Some of these parameters such as user's search history, popularity of a page (in terms of total visits) etc are not relevant to the context of ranking RDF triples. Web query results are typically large in size and also represent an approximate answer to the query as the query is imprecise. In contrast to this, SPARQL queries are precise and the results are typically smaller compared to web query results. In the semantic web context, consensus or agreement between data sources is the most relevant parameter and typically there are few links in the underlying graph that reflect consensus and hence contribute to result ranking. Thus we believe that RDF triple ranking requires a separate approach and web query ranking algorithms can not directly be used in this context.

3 System overview and Architecture

In this section, we describe the three layer model for ranking SPARQL query results. Fig. 1 shows the architecture of model, the top layer is called Dataset layer, middle layer is called Resource layer and the bottom most layer is called Triple layer.

Let D_i and D_j be two datasets that have similar entities. Here, by similar entities, we mean entities that represent same real world entity but have different names in different datasets. For example Tim-Berners Lee in DBpedia represented as `http://dbpedia.org/page/Tim_Berners-Lee` and in DBLP represented as `http://www4.wiwiss.fu-berlin.de/dblp/page/person/100007`. Say, D_i is well connected with other datasets in Linked Data cloud, but on the other hand, D_j is not well connected with other datasets. When similar entities from D_i and D_j appear in a query result, it is desirable to rank the entities of D_i higher than those for D_j. The intuition behind this is that by linkage to D_i, several independent data publishers have endorsed the data in D_i.

The dataset layer makes use of the Linked Data cloud graph[14] (Fig. 3) to assign ranking score to each dataset. A directional link from dataset D_i to D_j in LOD means that there are at-least 50 triples having their subject resource in D_i and object resource in D_j. A similar two-layer approach is used by DING[3], but

[14]http://richard.cyganiak.de/2007/10/lod/imagemap.html

the dataset score is calculated as a function of *number of inter-dataset links*. Since, the data is huge and number of links keeps increasing each day, dataset rank needs to be updated, each time some update is done in the dataset. This method becomes a very costly and time consuming, so here we propose a simpler way to calculate dataset rank using Linked Data cloud. The calculation of ranking score for datasets will be discussed in section 5.1.

The middle layer calculates ranking score for resources. The dataset layer takes into account all types of inter-dataset links, while the calculation of ranking score for resources is based only upon inter-dataset *owl:sameAs* links. Next section will discuss about the characteristic of *owl:sameAs* links and why only these links are used for calculating score for resources. The links shown in Fig. 1 at the resource layer are the inter-dataset *owl:sameAs* links.

The bottom most layer assigns a score to each triple of dataset, which is calculated using the scores of individual resources.

4 Ranking framework

The proposed ranking framework relies on the fact that, Linked Data published on the web obeys the principles of publishing Linked Data[15]. Our focus is not to calculate ranking score for new resources or resources specified only in one dataset. Rather, we propose a solution for resources which are specified in two or more datasets. However, the algorithm calculates ranking score for all the resources present in Linked Data cloud. The proposed method uses *owl:sameAs* links existing between resources to calculate ranking score. We studied practical use of *owl:sameAs* relationship and it appears best to use these links to model ranking score. The reason is that existence of *owl:sameAs* predicate between a pair of entities from two datasets indicates agreement between dataset creators that the two entities are same even though they have different URIs.

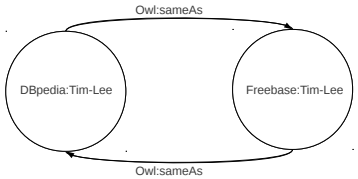

Figure 2: sameAs relationship for resource Tim-Lee

4.1 Practical use of *owl:sameAs*

The property *owl:sameAs* is defined as "the two URI's connected by *owl:sameAs* actually refer to same object". When a real world object is referred by two or more URI's, these URI's are called as URI aliases and *owl:sameAs* is used to connect these aliases. For example Tim-Lee in DBpedia is defined as `http://dbpedia.org/page/Tim_Berners-Lee` and in DBLP he is defined as `http://www4.wiwiss.fu-berlin.de/dblp/page/person/100007` and DBpedia has a *owl:sameAs* link from the former to the latter, so these two URIs are aliases of each other.

Though *owl:sameAs* is one of the most widely used predicates for connecting resources, the practical application of *owl:sameAs* is much different from how it is defined. The predicate *owl:sameAs* is transitive and symmetric, but in practice these characteristics are often violated. For example both Freebase[16] and Opencyc[17]have the concepts of football (a sport) and football players (person who plays this sport). Interestingly *football*[18] resource of Freebase is connected to *football player*[19] in Opencyc by *owl:sameAs* predicate. This is obviously incorrect and can never be symmetric. Though there are several instances of *owl:sameAs* relationships among Linked Data, not all of them are correct. The characteristic of symmetry is a minimal requirement for all *owl:sameAs* link to be correct. But we cannot deny the fact that there exist some unidirectional *owl:sameAs* links which are equally important as these bidirectional links. In the following section we make use of symmetric *owl:sameAs* instances for setting up the ranking framework.

Using *owl:sameAs*, a dataset states that a particular resource is same as another resource, that exists in a different dataset. If the other dataset also connects the same resource with *owl:sameAs* property to the first resource, it is most likely that both resources are same. For example, as shown in Fig. 2, the resource Tim-Lee in DBpedia is declared same as resource Tim-Lee in Freebase, and in addition Freebase confirms it back to DBpedia.

There are three possible types of links between two resources, which can create problems; they are explained in the following sections.

4.1.1 *sameAs* link exists but resources are different

This problem occurs when there is a unidirectional *owl:sameAs* link existing between two resources, but the resources are different. This may occur because of some malicious users trying to corrupt data. Since the data is open, a malicious user can add *owl:sameAs* link from own dataset's resource to well known resources of other datasets. Our framework is able to detect such

[15] `http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/`

[16] `http://www.freebase.com`

[17] `http://www.opencyc.org/`

[18] `http://www.freebase.com/view/en/football`

[19] `http://sw.opencyc.org/concept/Mx4rvVkDmZwpEbGdrcN5Y29ycA`

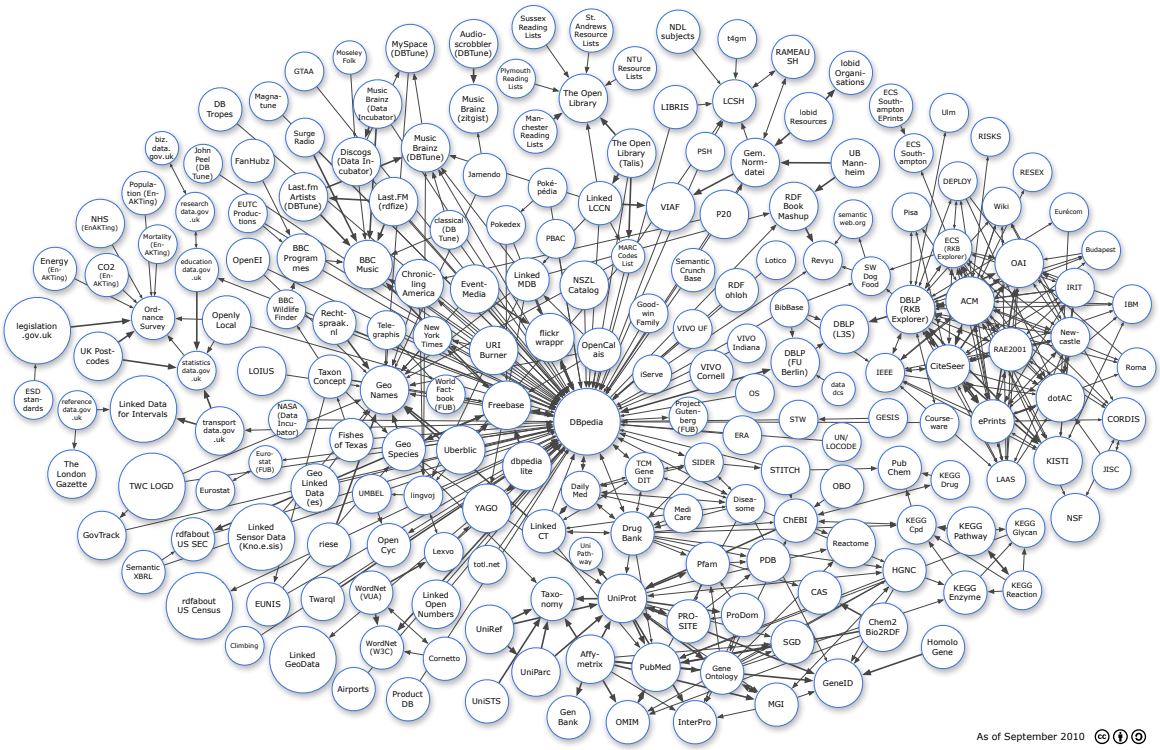

Figure 3: Linked Data cloud

possibilities and assign low ranking score for these resources.

4.1.2 *sameAs* link exists and resources are approximately same

This case arises when resources are slightly different in the way they are defined in two datasets. For example in Opencyc Sodium is defined in its pure form, while in DBpedia it is defined to include isotopes also. Obviously they are not same but, there exist a *owl:sameAs* link from DBpedia to Opencyc for this particular resource [7].

4.1.3 No *sameAs* link exists, but resources are same

It is extremely hard for dataset creators to regularly update their existing data. When their data is added to cloud they may not find any equivalent resource existing in other datasets and created a new resource. Since the same resource is defined in more than one dataset without *owl:sameAs* link, this is stated as problem of missing link. There exist some well known link discovery frameworks which identify such missing links, for example SILK[9].

5 Calculation of ranking score

The calculation is divided into two phases,

1. Dataset rank computation.
2. Triple rank computation.

5.1 Ranking score of a dataset

According to principles of publishing web of link data "A new dataset must always link to at least one of the existing dataset". This statement opens a way to rank the datasets. Since a new dataset has to depend on existing dataset, automatically an amount of belief is introduced into the cloud. A new dataset creator always tries to connect to information source that is having similar and trust-able data as of his/her own dataset. As more number of new datasets connect to a dataset, more will be the importance of that dataset, also as the ranking value of a dataset increases, more new datasets will try to connect to that dataset. Consider the LOD cloud[20] in Fig. 3, DBpedia seems to be the most important and popular dataset amongst all other datasets of cloud. And as the number of datasets linking to DBpedia increases, it is reasonable to assume that the value of DBpedia also tends to increase, which leads to higher rank of the particular dataset relative to others.
We make use of the statistics[21] published about

[20]Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. `http://lod-cloud.net`

[21]`http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/LinkStatistics`

Linked Data cloud for all calculations described below. In the proposed model, as the number of incoming links to a dataset increases, rank of that particular dataset also increases, which leads to higher rank. Since only one link[22] can be possible between any two datasets in LOD cloud, maximum number of possible incoming links is one less than the total number of datasets. So the ranking score for a dataset d, denoted as $R_{ds}(d)$, can be defined as:

$$R_{ds}(d) = \frac{\text{Total number of incoming links to dataset } d}{\text{Total number of datasets in LOD cloud}}$$

In the next section, we explain how the ranking score of datasets are used to calculate ranking score of individual resources and triples.

5.2 Ranking score of resources and triples

The ranking score of a triple is defined in terms of the ranking score of individual resources. Ranking score of a resource is calculated using *owl:sameAs* relation existing between two datasets. Here the practical uses of *owl:sameAs* will be exploited, as described in Section 4.1. Three types of *owl:sameAs* links are possible:

Type 1. Outgoing link to a resource in a different dataset.
Type 2. Incoming link from a resource from a different dataset.
Type 3. Incoming and outgoing links between a pair of resources, where both resources are from different datasets.

From these three types of links, we propose to use only the last two types of links for calculation of ranking score of resources. Outgoing *owl:sameAs* links from a resource to other resources in different datasets can not be used for rank calculation. This is because, a new dataset may claim that a resource it introduces is *owl:sameAs* many others but unless other dataset owners confirm the relationship, it is not trustworthy. However, the incoming *owl:sameAs* links can be used to calculate ranking score as an independent dataset owner has asserted this relationship. The third type of links that are bidirectional are the ones that reflect mutual confidence between a pair of data sources and can certainly be used for calculation. We call the values calculated from the second type of links as *partial score* values and values calculated from the third type of links as *mutual score* values.

Fig. 4 shows the existing *owl:sameAs* links for the DBpedia resource Tim-Lee, among the shown datasets. The number inside the nodes represents the type of link discussed above.

[22] http://richard.cyganiak.de/2007/10/lod/

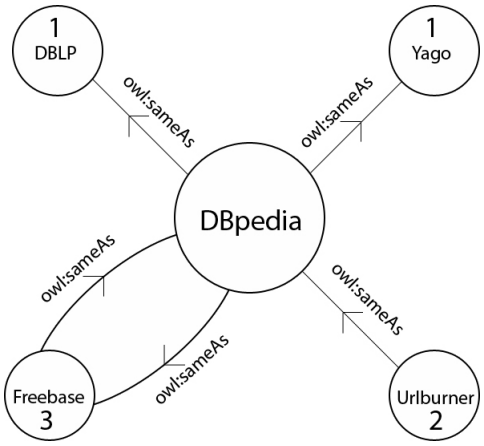

Figure 4: *owl:sameAs* graph for resource Tim-Lee

5.2.1 Mutual score

Mutual score is calculated using Type 3 links, discussed in previous section. Let r_1, r_2,...., r_n be the resources that exist in different datasets and have Type 3 links to resource r in the dataset d. Now, mutual score for a resource r, denoted as $R_{mutual}(r)$, is defined as:

$$R_{mutual}(r) = \sum_{i=1}^{n} R_{ds}(\text{dataSetOf}(r_i))$$

Here $dataSetOf(r_i)$ denotes the dataset, say d_i, that contains resource r_i.

5.2.2 Partial score

Partial score uses Type 2 links for calculation. Let r be a resource which has an incoming *owl:sameAs* link from resource r_k. Also, let p_k be number of outgoing and non-bidirectional links from r_k to other resources. Then r_k will transfer $R_{ds}(\text{dataSetOf}(r_k))/p_k$ amount to resource r. Partial score of a resource r, denoted as R_{pa} is defined as:

$$R_{pa}(r) = \sum_{k=1}^{m} \frac{R_{ds}(\text{dataSetOf}(r_k))}{p_k}$$

m = Number of resources, each from different datasets having type 2 links to r.

Partial score mainly takes care of link spamming. Intuitively a unidirectional *owl:sameAs* link is of less use compared as a bidirectional owl:sameAs link. However if the dataset that has several out-going *owl:sameAs* links is indeed authentic, over a period of time the links become bidirectional and the rank of the resource also increases.

5.2.3 Total resource ranking score

The ranking score of a resource, denoted as $T(r)$, is defined as[23]:

$$T(r) = R_{mutual}(r) + R_{pa}(r)$$

Theoretically, a *owl:sameAs* network for a resource can be designed which can make ranking of a resource greater than 1. But practically, the cloud is so distributed that score value of a resource hardly goes beyond 1. In case, the value goes beyond 1, we truncate it to 1. Thus, the score value of a resource is in the range [0,1].

5.2.4 Triple ranking score

A triple is defined as subject - predicate - object. Where each of the three refers to a resource or a concept in real world. Score of a triple is calculated by adding score values of individual resources of triple. The *triple score* of a triple, denoted as R_{triple}, is defined as:

$$R_{triple} = (T(r_{subject}) + T(r_{predicate}) + T(r_{object}))/3$$

Since score of a resource is in the range of [0,1], adding such three resource score can reach maximum limit of 3. To normalize score value of triple we divided the score by 3. The Triple score of a triple is in the range of [0,1].

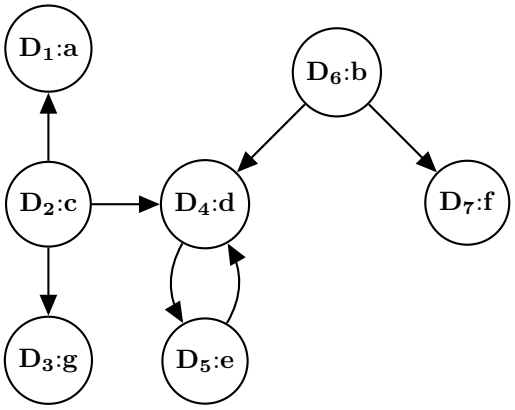

Figure 5: sameAs network for node d of dataset D_4

5.3 Example

Fig. 5 shows an example *owl:sameAs* network for resource d in dataset D_4. Here D_4:d represents a resource d of dataset D_4. Now, calculating score for resource d :

[23] $T(r)$ value of known resources such as *owl:inverseOf*, *rdf:seeAlso* and others defined by W3C and xmlns community is taken as 1. Because these resources are most valuable resources.

Since D_5:e is the only resource having Type 3 link to D_4:d, also, D_2:b and D_6:c are the resources having Type 2 links to D_4:d, with 2 and 3 unidirectional outgoing links respectively,

$$R_{mutual}(d) = R_{ds}(\text{dataSetOf}(e))$$

$$R_{pa}(d) = \frac{R_{ds}(\text{dataSetOf}(b))}{2} + \frac{R_{ds}(\text{dataSetOf}(c))}{3}$$

$$T(d) = R_{mutual}(d) + R_{pa}(d)$$

After calculation, score can be attached to particular triple and can be represented in form of quads in place of triples. Here fourth attribute represents ranking score.

6 Algorithm

Our algorithm is divided into three parts, Algorithm 1 calculates dataset score for all the datasets and puts them in a list, Algorithm 2 calculates resource score for all resources and puts them in a dictionary with resource as key and score as value, Algorithm 3 calculates triple score for all triples and writes them to a file. In addition, we define, BTC cloud(will be discussed in section 7) as the subset of LOD cloud that contains datasets common in BTC dataset and LOD. The input for these algorithms are:

- A file containing all resources
- A file containing all triples
- A file containing triples with *owl:sameAs* link only

Algorithm 1: Dataset-score(r)

Input : A Resource r
Output: Dataset score of r
1 d = dataSetOf(r)
2 In = Number of incoming links to Dataset d from other datasets in BTC cloud.
3 Total = Total number of datasets in BTC cloud
4 $R_{ds} = \frac{In}{Total}$
5 return R_{ds}

Algorithm 1 takes input as a resource r then it calls for function dataSetOf(r) to find dataset d from which this resource belongs to and calculate number of incoming links to dataset d. Line 4 calculates dataset score for dataset d.

Algorithm 2 takes a resource r as an input. Line 1 initialize *listIn*, *listBoth* and *listPartial* lists. Here *lineIn* stores resources having outgoing *owl:sameAs* link to resource r, *listBoth* stores resources having bidirectional *owl:sameAs* link with resource r and *listPartial* stores unidirectional outgoing *owl:sameAs* link of resources in *listIN* at time of calculating partial score. Lines 2-3 find resources with outgoing link

Algorithm 2: Resource-score(r)

Input : A resource r from list of resources *ResourceList*
Output: Resource-score of r

1 $listIN \leftarrow null$, $listBoth \leftarrow null$, $listPartial \leftarrow null$
2 Find resources r_i with incoming link to r
3 Append r_i to list $listIN$
4 **foreach** $r_{in} \in listIN$ **do**
5 | Find resources r_j with incoming link to r_{in}
6 | **if** $r_{in} = r$ **then**
7 | | Add to list $listBoth$
8 | **end**
9 **end**
10 $listIN = listIN - listBoth$
11 $R_{mutual} \leftarrow 0$, $R_{pa} \leftarrow 0$, $T \leftarrow 0$
12 **if** $listBoth = null$ **then**
13 | $R_{mutual} = 0$
14 **end**
15 **else**
16 | **foreach** $r_i \in listBoth$ **do**
17 | | $R_{mutual} = R_{mutual} + getDataSet\text{-}score(r_i)$
18 | **end**
19 **end**
20 **foreach** $r_i \in listIN$ **do**
21 | $count \leftarrow 0$, $Res \leftarrow null$
22 | Find resources Res with outgoing link to r_i
23 | **foreach** $r_o \in Res$ **do**
24 | | **if** $\notin$ *outgoing link from* r_o *to* r_i **then**
25 | | | $count = count + 1$
26 | | **end**
27 | **end**
28 | $R_{pa} = R_{pa} + getDataSet\text{-}score(r_i)/count$
29 **end**
30 $T = R_{mutual} + R_{pa}$
31 return T

Algorithm 3: Triple-score(t)

Input : A Triple t
Output: Triple-score of t

1 $R_{triple} = (getResource\text{-}score(r_{subject}) + getResource\text{-}score(r_{predicate}) + getResource\text{-}score(r_{object}))/3$
2 return R_{triple}

to r and append it to $listIN$. Lines 4-9 find resources having bidirectional *owl:sameAs* links with r and appends that to $listBoth$. Now, there are some resources which will be in both the list $listIN$ and $listBoth$, so, line 10 takes both lists and remove resources from $listIN$, which are already in $listBoth$. Now, $listIN$ contains resources which have only outgoing link to r and not having bidirectional links. $listBoth$ now contains type-3 resources and $listIN$ contains type-2 resources. Lines 12-19 check if $listBoth$ is empty, if it is empty then mutual score (R_{mutual}) will be 0. Otherwise, use getDataset-score module to get dataset-score of r_i and calculate R_{mutual}. Lines 20-27 take a resource r_i from $listIN$ and count for number of outgoing unidirectional *owl:sameAs* links to other resources. Line 28 calculates partial score for each r_i and sum for all r_i to produce total partial score. Line 30 sums up mutual score and partial score of r to calculate resource score.

Algorithm 3 takes a triple from a file containing all triples as input. At line 1, algorithm calls for getResource-score module, to get resource score of subject, predicate and object resources from dictionary containing resource score. By adding value returned, triple score value of a triple is obtained.

7 Experimental setup and Results

We have chosen BTC 2010[24] dataset for experimental verification of proposed framework. BTC dataset is one of the largest datasets available, it consists of 3.2 Billion Triples. The dataset consists of data crawled from several other datasets where each resource is represented by a URI. We mapped these URIs with their domain names and found a total of 1132 domain names. Since our experiment depends on LOD cloud, we found there are 55 such datasets in common between LOD cloud and BTC dataset. From these 55 datasets we found only 31 dataset's resources having links within the set of 55 datasets, all other 24 data sets having all the links out of these 55 datasets. So, we performed our experiment on these 31 dataset resources present in BTC data. We call this newly formed dataset as BTC-cloud dataset.

We performed our experiments on 4 machines, all running Ubuntu 10.10 with Quad-core 2.83GHZ processor having 4 GB RAM and 2TB of Hard disk. Most of the work in this experiment is done using Allegrograph 4.2[25]. Since the data is large, we parallelized execution by using one machine running Allegrograph server and rest of the machines as clients.

We performed the following series of operations on BTC data to get it in required form:

- Convert N-Quads to N-Triples format
- Removed unwanted triples and triples containing literals
- Mapped resources to their domain names
- Found common datasets between LOD cloud and BTC
- Divided BTC-cloud dataset into 31 sub-datasets for calculating score

Since our algorithm is based on *owl:sameAs* network, our next operation is to extract triples having *owl:sameAs* relationship existing between resources of different datasets. The *owl:sameAs* network formed is used for calculation of ranking score.

[24] http://km.aifb.kit.edu/projects/btc-2010
[25] `http://www.franz.com/agraph/allegrograph/`

Dataset	DS-score	Dataset	DS-score	Dataset	DS-score	Dataset	DS-score
DBLP (rkb)	0	Musicbrainz	0.1290	DBpedia	0.5806	OS (rkb)	0
Crunch Base	0	Dailymed	0.0967	Linkedct	0.0967	Diseasome	0.0967
Drugbank	0.1290	DBLP Berlin	0.0645	DBLP Hannover	0.0645	Freebase	0.0645
Lingvoj	0.0645	SW Conference Corpus	0.0645	Eurostat	0.0322	Factbook	0.0645
Project Gutenberg	0.0322	Revyu	0.0645	Jamendo	0.0645	Linkedmdb	0.0967
Yago	0.1290	Myspace	0.0322	Surgeradio	0.0322	Opencalais	0
Opencyc	0.0645	Umbel	0.0322	Wikicompany	0.0322	Openguides	0.0322
telegraphis (capitals)	0.0322	telegraphis (countries)	0.0322	Geonames	0.2580		

Figure 6: Dataset-score of all 31 datasets used in the experiment

Resource(Dihydrofolate Reductase)	Ranking score
http://www4.wiwiss.fu-berlin.de/drugbank/resource/targets/365	0.2903225805
http://dbpedia.org/resource/Dihydrofolate_reductase	0.129032258
http://mpii.de/yago/resource/Dihydrofolate_reductase	0.0

Table A

Resource(Apple Island)	Ranking score
http://sws.geonames.org/4984314	0.580645161
http://dbpedia.org/resource/Apple_Island	0.258064516

Table B

Figure 7: Resource-score

After these series of operations we found that there are approximately 1.8 million unique resources, more than 10 million unique triples and about 1 million triples with *owl:sameAs* predicate.

Our goal in this work is to rank SPARQL query results, but the ranking scores of datasets and resources produced during the process can also be used for ranking datasets and resources.

The ranking score of datasets is shown in Fig. 6. Here DBpedia is having highest score. The resource ranking score produced can be used for ranking entities in entity search engines. The ranking score produced are domain independent and indicates the global popularity of individual. Fig. 7 shows two sets of resources and their ranking score. here, all entities in a set represent the same real world entity. Table A shows the entity named "Dihydrofolate reductase" and its representation in three different datasets, namely DBpedia and YAGO[26] that are cross-domain datasets and drugbank[27], a biology dataset. Here point to notice is that even though the dataset rank of DBpedia is higher, the score of DBpedia resource is less than Drugbank. A similar example in Table B shows the entity named "Apple Island" and its representation in two datasets, Geonames and DBpedia, where Geonames[28] is geographical dataset. Here also the entity from the relevant domain dataset gets higher score than the other.

Fig. 8 shows how the results of a SPARQL query would get ordered by the ranks of the triples. The query is as below:

select ?s where {
?s <rdf:type> <dbpediac:LandscapeArtist>}[29]

8 Scalability

The latest statistics about web of data shows the amount of open Linked Data available on the web[30]. The algorithm in section 6 assumes that the data is available locally. Clearly, this approach does not work for actual LOD as the size of the data is very large. However, in order to calculate the ranking scores proposed in this paper, we need only the triples that have *owl:sameAs* predicate. The subgraph of LOD graph consisting of these links is much smaller and can be stored locally. Thus the ranking scores of resources can be computed locally whereas scores for triples can be computed on demand whenever they are required to be reported as results. A better strategy for computing and using the ranking scores needs to be worked out as part of future work.

9 Conclusion and Future Work

A key difference between Linked Data and other data sharing approach is, its open nature. Linked Data works on the principle of "anyone can say anything about anything". Till now, the number of datasets in Linked Data cloud gets doubled every year and is attracting a lot of interest from researchers and commercial organization.

[26]http://www.mpi-inf.mpg.de/yago-naga/yago/
[27]http://www4.wiwiss.fu-berlin.de/drugbank/
[28]http://www.geonames.org

[29]dbpediac : <http://dbpedia.org/class/yago/>
rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
dbpediar : <http://dbpedia.org/resource/>
[30]http://www4.wiwiss.fu-berlin.de/lodcloud/state/

Subject (?s)	Predicate	Object	Triple score
dbpediar:Charles_Leickert	rdf:type	dbpediac:LandscapeArtists	0.397849462
dbpediar:Bernard_Hailstone	rdf:type	dbpediac:LandscapeArtists	0.376344086
dbpediar:John_Marin	rdf:type	dbpediac:LandscapeArtists	0.376344086
dbpediar:Lucius_Richard_OBrien	rdf:type	dbpediac:LandscapeArtists	0.333333333

Figure 8: Answer(?s) of the SPARQL query with Triple-score

In this paper, we have proposed a framework for ranking entities as well as triples. For this we studied practical usage of *owl:sameAs* predicate and how it's use being violated while designing ontologies and RDF datasets. We used our findings to calculate score of triples, and stored this data in the form of N-Quads.

The current framework is limited to ranking individuals because *owl:sameAs* can not be used between predicate resources. Future work includes extending the current framework to rank predicate resources using vocabularies and extending the functionality of SPARQL processor to query the data produced in form of N-Quads. Also, if a resource occurs in multiple datasets, the link structure between these resources can be unidirectional or bidirectional. In case of unidirectional links there is possibility of either removing the link if it is not valid or converting it to a bidirectional link. Link discovery in linked data is a concern and several research efforts are going on in this direction. Considering the huge size of the evolving linked data, this becomes a difficult problem. Our algorithm makes use of links available in datasets, there is possibility of explicitly discovering *owl:sameAs* links using available framework like SILK [9]. Then we can use this information for our calculation. This paper does not consider transitive nature of *owl:sameAs* relationships. It would be interesting to to extend the approach to use these relationships.

References

[1] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, WWW, 2005.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[3] R. Delbru, N. Toupikov, M. Catasta, G. Tummarello, and S. Decker. Hierarchical link analysis for ranking web data. In *Proceedings of European Semantic Web Symposium / Conference*, 2010.

[4] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM, 2004.

[5] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and Ranking Knowledge on the Semantic Web. In *Proceedings of the 4th International Semantic Web Conference*, 2005.

[6] T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *Proceedings of International Semantic Web Conference*, 2009.

[7] H. Halpin, P. Hayes, J. P. McCusker, D. McGuinness, and H. S. Thompson. When owl:sameAs isn't the same: An analysis of identity in Linked Data. *In Proceedings of 9th International Semantic Web Conference*, 2010.

[8] A. Hogan, A. Harth, and S. Decker. Reconrank: A scalable ranking method for semantic web data with context. In *Proceedings of 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[9] A. Jentzsch, R. Isele, and C. Bizer. Silk - Generating RDF links while publishing or consuming Linked Data. *In Proceedings of 9th International Semantic Web Conference*, 2010.

[10] Z. Jing, M. Chune, Z. Chenting, Z. Jun, Y. Li, and M. Xinsheng. A novel ranking framework for linked data from relational databases. *TSINGHUA SCIENCE AND TECHNOLOGY*, 2010.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, September 1999.

[12] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, WWW, 2005.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford Digital Library Technologies Project*, 1998.