VU VRIJE UNIVERSITEIT AMSTERDAM

VU Research Portal

Finding the Achilles Heel of the Web of Data : using network analysis for link-recommendation.

Gueret, C.D.M.; Groth, P.T.; van Harmelen, F.A.H.; Schlobach, S.

published in
Lecture Notes in Computer Science
2010

DOI (link to publisher)
10.1007/978-3-642-17746-0_19

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)
Gueret, C. D. M., Groth, P. T., van Harmelen, F. A. H., & Schlobach, S. (2010). Finding the Achilles Heel of the Web of Data : using network analysis for link-recommendation. *Lecture Notes in Computer Science*, *6496*. https://doi.org/10.1007/978-3-642-17746-0_19

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 21. Jan. 2025

Finding the Achilles Heel of the Web of Data: Using Network Analysis for Link-Recommendation

Christophe Guéret, Paul Groth, Frank van Harmelen, and Stefan Schlobach

VU University Amsterdam
De Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands
{cgueret,pgroth,Frank.van.Harmelen,schlobac}@few.vu.nl

Abstract. The Web of Data is increasingly becoming an important infrastructure for such diverse sectors as entertainment, government, e-commerce and science. As a result, the *robustness* of this Web of Data is now crucial. Prior studies show that the Web of Data is strongly dependent on a small number of central hubs, making it highly vulnerable to single points of failure. In this paper, we present concepts and algorithms to analyse and repair the brittleness of the Web of Data. We apply these on a substantial subset of it, the 2010 Billion Triple Challenge dataset. We first distinguish the *physical structure* of the Web of Data from its *semantic structure*. For both of these structures, we then calculate their robustness, taking *betweenness centrality* as a robustness-measure. To the best of our knowledge, this is the first time that such robustness-indicators have been calculated for the Web of Data. Finally, we determine which links should be added to the Web of Data in order to *improve* its robustness most effectively. We are able to determine such links by interpreting the question as a very large optimisation problem and deploying an evolutionary algorithm to solve this problem. We believe that with this work, we offer an effective method to analyse and improve the most important structure that the Semantic Web community has constructed to date.

1 Introduction

The rapidly growing Web of Data increasingly resembles the Web in its network properties. It resembles a small world network that relies on central hubs to provide connectivity between resources on the Web of Data [10]. Such central hubs are potential points of failure. This is particularly dangerous for the Web of Data, which, unlike the Web, is designed to be used by automated agents that have less capability to recover from lack of access to resources than human users might have on the regular Web.

Current approaches to ensure robustness of the Web of Data are based on anecdotal observations. In this work, we propose a systematic approach for analysing the Web of Data and recommending where links can be added to

P.F. Patel-Schneider et al. (Eds.): ISWC 2010, Part I, LNCS 6496, pp. 289–304, 2010.
© Springer-Verlag Berlin Heidelberg 2010

help ensure robustness against both infrastructure failure and semantic deviation. An example of the first is: how can we ensure that automated agents can still traverse the network if DBpedia is down? An example of the second is: if the SIOC ontology is updated, where do links need to be introduced to re-establish connectivity?

Our systematic approach uses well known network properties to characterise the robustness of both the infrastructure and semantic networks within the Web of Data. Based on these properties, we present an optimisation algorithm that produces recommendations about where links should be added to the Web of Data. The algorithm takes into account whether additional links would be semantically meaningful.

The contributions of this paper are (i) a characterisation of the strength of the current Web of Data in terms of its infrastructure and semantic network.; (ii) a recommendation algorithm for adding links to the Web of Data to increase its robustness; and (iii) applying this algorithm in order to determine how many (and which) links are required to obtain different levels of robustness.

Our main findings are that (a) the current Web of Data is indeed highly sensitive to failure of individual nodes, both at the infrastructure level and as a semantic network, and (b) this situation can be remedied by adding a surprisingly small number of links, provided that these links are chosen well, as calculated by our recommendation algorithm.

The paper is organised as follows. In Section 2, we discuss related work and argue why it is useful to distinguish infrastructural connectivity and semantic connectivity. This leads to Section 3 where the robustness of the current Web of Data is measured, followed by Section 4, which presents an algorithm to recommend how best to increase that robustness. Section 5 concludes.

2 Background

2.1 Related Work

The use of network properties to study complex systems has grown in a wide range of fields (*e.g.* biology, social science and web science) because it provides a mechanism to extract global properties of systems [12]. In terms of robustness, the classic result is from Barabasi, which shows that scale-free networks are robust against random failure, but not against targeted attacks [1]. The robustness of scale-free networks is important because they are widely seen in nature including power grids, the World Wide Web and social networks [2].

The application of such network analysis to the Web of Data has until now been limited, and has been performed on a wide variety of graph-structures: [10] analysed the 2009 BTC dataset[1] and showed that, interpreted as a sample of the Web of Data, it is scale-free and that `semanticdesktop.org` and `purl.org` are central in it. The same paper also analysed the well-known "bubble-graph" of the Web of Data, consisting of the datasets published and interlinked by the

[1] `http://vmlion25.deri.ie/index.html`

Linking Open Data project[2]. It showed the existence of topic-oriented hubs, with DBPedia connected to 50% of all the other datasets, and over 50% of all the shortest paths in the graph being routed through either DBPedia or DBLP.

In recent work, [8] analysed the "object link graph": the Web of Data restricted to its object-to-object links, i.e. after removing all links from objects to classes, and all class- and property-hierarchies. They found that this object-link graph also has a scale-free nature, with a diameter value of 12, which is small compared to the size of the graph, although the link density is rather low. Such a small diameter of a large but low density graph again points to the presence of central hubs that provide the main connectivity between many resources.

Other work, such as [9], also use network analysis tools, but apply them only to networks of ontologies, and do not consider the much more substantial collection of instances that form the real content of the Web of Data. At an even smaller scale, [13] applies concepts from network analysis to individual ontologies.

Summarising, only a handful of analyses have been performed on the network properties of the Web of Data. Furthermore, all these works have only *analysed* the Web of Data, but nobody has used the results of their analysis to effectively compute *improvements* to the Web of Data.

2.2 Infrastructure Failure and Semantic Failure

Connectivity on the Web of Data can be disrupted in two different ways: infrastructural failure or semantic failure. For the infrastructure, the problem is server unavailability, *e.g.* the `dbpedia.org` server is down. In the semantic network, the problem is robustness against change, for example still using `sioc:User` instead of the current `sioc:UserAccount`.

The robustness of an infrastructure is commonly improved by the use of mirrors and caches. Our approach is complimentary to using these techniques. In order to detect hosts that function as infrastructure hubs, and whose unavailability would hence break many paths, we aggregate the Web of Data into a *hostname graph*:

Definition 1 (hostnames graph). *The hostname graph $\mathcal{H}$ is a $\langle V, E\rangle$ where $h \in V$ is a node of $\mathcal{H}$* iff *h is used as a hostname in any URI on the Web of Data, and $e \in E, e = \langle h_1, h_2\rangle$ is an edge of $\mathcal{H}$ from node h_1 to node h_2* iff *there is a triple $\langle s, p, o\rangle$ anywhere on the Web of data with h_1 the hostname referred to in the URI of s and h_2 the hostname referred to in the URI of o.*

Thus, the hostname graph has as many nodes as there are hostnames mentioned in all the triples on the Web of Data.

Similarly, the namespace graph is an aggregation of the semantic structure of the Web of Data:

Definition 2 (namespaces graph). *The namespace graph $\mathcal{S}$ is a tuple $\langle V, E\rangle$ where $n \in V$ is a node of $\mathcal{S}$* iff *n is used as a namespace anywhere on the Web of*

[2] `http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

Data, and $e \in E, e = \langle n_1, n_2 \rangle$ *is an edge of* $\mathcal{S}$ *from node* n_1 *to node* n_2 iff *there is a* $\langle s, p, o \rangle$ *anywhere on the Web of Data with* n_1 *the namespace of* s *and* n_2 *the namespace of* o.

Thus, the namespace graph has as many nodes as there are namespaces mentioned in all the triples on the Web of Data.

Definition 3 (content of nodes). *The content* $cont(n)$ *of a node* n *is defined as the set of URI such that there is a* $\langle r, p, o \rangle$ *anywhere on the Web of Data and* n *is the namespace of* r *for a namespaces graph or* n *is the hostname of* r *for an hostnames graph.*

3 Analysing the Web of Data

The networks and the programs described in this section are all publicly available at `http://linkeddata.few.vu.nl/wod_analysis/`.

3.1 Measures of Robustness

By robustness of a graph, we mean the degree to which connectivity in a graph is maintained after a node is removed from the graph. There are a number of network measures that can be used for measuring the robustness of a graph. For example, the diameter of a graph[3] provides information about connectivity. A smaller diameter implies that there are a large number of connections within the network while a larger diameter means that the network is less connected. While the diameter provides a reasonable global summary statistic, centrality statistics allow one to investigate the graph on a per node basis. In particular, betweenness centrality measures how often a node occurs on a shortest path any pair of nodes:

Definition 4 (Betweenness centrality). *For a graph* $G = (N, E)$ *with a set of nodes* N *and a set of edges* E, *the* betweenness centrality $B(n)$ *of a node* $n \in N$ *is defined as*

$$B(n) = \sum_{s \neq n \neq t \in N} \frac{S(s,n,t)}{S(s,t)}$$

where $S(s,t)$ *is the number of shortest paths from* s *to* t, *and* $S(s,n,t)$ *is the number of shortest paths from* s *to* t *that pass through node* n. *Instead of* $B(n)$ *we will often report on its* non-normalised *version,* $B'(n)$.

$$B'(n) = \sum_{s \neq n \neq t \in N} S(s,n,t),$$

Instead of "betweenness centrality" we will often simply speak of "betweenness".

[3] The diameter of a graph is the longest shortest path in the graph.

Betweenness is a measure of the importance of a node for the connectivity between other nodes. The intuition is that if a node lies on many shortest paths it is an important node, since removal of such a node will directly influence the cost of the connectivity between other nodes, as other (i.e., longer) shortest paths will have to be followed.[4]

A completely connected network has the maximal robustness, and correspondingly the lowest betweenness centrality: $B(n) = 0$ for every $n \in N$, and removing one node does not impact the overall connectivity of the network greatly.

If we want to improve the robustness of the Web of Data, we will want to lower the number of nodes that have high betweenness centrality, since these are important potential points of failure. For this, we will first need to analyse which nodes actually have a high betweenness centrality. This is obviously computationally intensive, since it involves calculating the shortest paths between all pairs of nodes on the Web of Data. This *robustness analysis* will be topic of the remainder of this section. Deciding how to *improve the robustness* will be tackled in Section 4.

3.2 Dataset

The 2010 Billion Triple Challenge (BTC) Dataset[5] was used as a representative sample of the Web of Data. It contains roughly 3.2 billion statements. From this dataset, the hostname graph and namespace graph were constructed. Given that namespaces cannot be systematically identified given a URL alone, we used a predefined list of widely used namespaces as defined by the `prefix.cc` service. Out of the 330 namespaces registered on the services, 198 were found to be used in the snapshot used to create the networks.

We removed from the BTC all triples where the object was a literal, all triples containing blank nodes, and all triples that refer only to URI's from the same dataset, since none of these triples would contribute to the objects of our study, namely the hostname graph and the namespace graph. Surprisingly, this reduced the BTC dataset to 530 million triples, showing that the vast majority of the Web of Data (or at least the BTC snapshot of it) does not contribute to it being a "web". Of those remaining 530 million triples, the vast majority (389 million) were covered by the namespace list built from `prefix.cc`. This gives us some confidence that the namespace list is sufficiently representative set of namespaces for building our namespace graph.

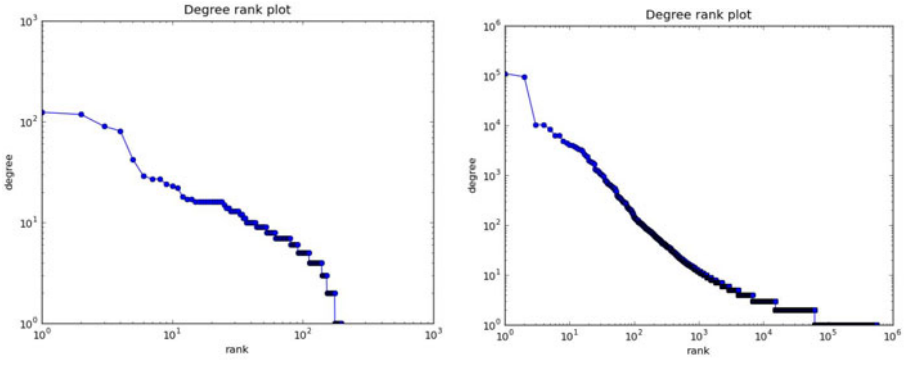
As a further characterisation of our dataset, Figure 1 shows the degree distribution of both the hostname graph (infrastructure) and the namespace graph (semantic links). Both distributions exhibit a pattern that is not linear. The

[4] Of course, if we are interested in connectivity, it is only an approximation to assume that connections only happen along shortest paths; variations of betweenness centrality such as "flow betweenness" and "random walk betweenness" have been proposed to allow for this. In many practical cases however, the simple (shortest path) betweenness centrality gives quite informative answers [12].

[5] `http://km.aifb.kit.edu/projects/btc-2010/`

Table 1. Size of the two studied networks

Network name	Number of nodes	Number of edges
Hostnames	558841	656012
Namespaces	198	936



(a) Namespaces (b) Hostnames

Fig. 1. Degree distribution of the namespaces and hostnames networks

degree shown in distribution does not follow a power law. From this we can conclude that that these two networks are not scale-free. However, they still have a few strongly connected hubs.

3.3 Robustness Results

Based on the extracted graphs, we calculated the betweenness centrality for all nodes in both graphs using the Small-world Network Analysis and Partitioning software (SNAP) [4]. Given the size of the hostname graph, we used an approximation algorithm implemented by SNAP and set the sampling percentage to 10% of all nodes. This is double the 5% percentage suggested for use in [4]. For more details on the algorithm used, see [3].

Infrastructure Analysis. Table 2(a) shows the non-normalised betweenness distribution ($B'(n)$) among the hostnames on the Web of Data, in ten bins starting from the maximal centrality and working down to zero. We note that the distribution does not follow a power-law curve but is in fact more extreme: essentially, almost all infrastructural connectivity on the Web of Data is mediated by only 3 servers. Table 2(b) reveals which hosts these are: `xmlns.org`, `dbpedia.org` and `purl.org`. All this points to an extreme brittleness of the infrastructure underlying the Web of Data: only taking out a handful of servers would completely cripple the entire network.

Figure 2 provides a good example of the potential impact that the dominance of hubs could have on the Web of Data. Recently, Radar Networks which owned `www.twine.com` was sold to another company Evri[6]. While the transition of `twine.com` to Evri was smooth, it is entirely possible that `www.twine.com` could have ceased to exist or no longer supported Web of Data content as a result of this takeover. Our analysis shows that this would have had a substantial impact on the infrastructural connectivity of the Web of Data.

Table 2. Histogram of betweenness for hostnames and the top ten hostnames with the highest betweenness

$B'(n)$[7]	**#Nodes**
$5 - 6 \times 10^9$	2
$4 - 5 \times 10^9$	0
$3 - 4 \times 10^9$	0
$2 - 3 \times 10^9$	1
$1 - 2 \times 10^9$	0
$0.5 - 1 \times 10^9$	4053
$0 - 0.5 \times 10^9$	554785

(a) Distribution of the betweenness results

Hostname	$B'(n)$
`xmlns.com`	5 693 379 049
`dbpedia.org`	5 432 125 038
`purl.org`	2 163 504 423
`www.kanzaki.com`	532 149 372
`www.w3.org`	470 113 796
`dbtune.org`	323 796 691
`identi.ca`	318 896 524
`www.twine.com`	299 237 555
`semanticweb.org`	277 374 029
`dblp.l3s.de`	225 602 575

(b) Top 10 hostnames and their betweenness result

The 554 785 hostnames with a betweenness of 0 are dead ends in the network. Some of these hosts may be used to serve only non semantic content, such as images. Thus, they do not provide resources that can be interlinked and used to walk through the network. The 4056 other hosts are more representative of the interlinkage status of the graph. This number is much higher than the 198 nodes in the namespaces network (these namespaces account for 60 different hostnames).

Semantic Network analysis. Similar to the infrastructure network analysis, Table 3a shows the betweenness distribution of the namespaces, again arranged in 10 bins. The majority of nodes are not in-between at all and the overall distribution mirrors that of the hostnames graph. The semantic network of the Web of Data, like its infrastructure network, also relies heavily on hubs. Table 3b shows these hubs. These are indeed the hubs one would expect, perhaps with the exception of `example.org`, which, by definition, can provide no connectivity to other namespaces because it is reserved for examples[8].

[6] `http://www.novaspivack.com/uncategorized/evri-ties-the-knot-with-twine`
[7] Non-normalised betweenness.
[8] See RFC2606, `http://www.rfc-editor.org/rfc/rfc2606.txt`

Table 3. Histogram of betweenness for namespaces and the top ten namespaces with the highest betweenness

$B'(n)$	#Nodes
8001-9000	1
7001-8000	1
6001-7000	0
5001-6000	2
4001-5000	0
3001-4000	1
2001-3000	0
1001-2000	6
1-1000	70
0	117

(a) Distribution of the betweenness results

Namespace	$B'(n)$
www.w3.org/1999/02/22-rdf-syntax-ns#	8783
example.org/	7191
dbpedia.org/resource/	5428
xmlns.com/foaf/0.1/	5030
www.w3.org/2002/07/owl#	3926
sw.opencyc.org/concept/	1764
www.w3.org/2007/uwa/ context/deliverycontext.owl#	1737
www.w3.org/2003/01/geo/wgs84_pos#	1609
www.semanticdesktop.org/ ontologies/2007/11/01/pimo#	1300
ontologies.ezweb. morfeo-project.org/eztag/ns#	1225

(b) Top 10 namespaces and their betweenness result

4 Improving the Web of Data

The previous section has shown that the Web of Data is extremely brittle, and relies on a very small number of hubs that are crucial to its connectivity. Both the infrastructure network and the semantic network could be be strengthened by judiciously adding links to the network The expected impact of such new links is to reduce the *variation of the centrality* among the nodes of a graph, thereby diminishing the importance of hubs. The variation of betweenness centrality within a graph is termed the *centralisation betweenness index* [7]:

Definition 5 (centralisation betweenness index). *Given a graph,* $G = (N, E)$ *with a set of nodes* N *and a set of edges* E*, the centralisation betweenness index* $C(G)$ *of* G *is defined as*

$$C(G) = \sum_{i=1}^{G} \frac{[max_{n \in N}(B(n)) - B(i)]}{(|N| - 1)}$$

where $B(n)$ *is the betweenness of node* n *in the graph.*

4.1 The Cost of Fixing the WoD

The simplest way of reducing $C(G)$ would be to make G a fully connected graph, resulting in an optimal value of $C(G) = 0$. Of course, for the Web of Data this is neither feasible nor desirable, because only semantically meaningful links should be added. Besides, the creation of new edges has a cost. As is well known from the ontology mapping domain, establishing new relations between two ontologies is no easy task. Similarly, finding equivalent instances that can be related by a sameAs triple is challenging.

We have therefore chosen to characterise the problem of recommending where to introduce edges in the Web of Data as an optimisation problem that minimises the centralisation index $C(G)$ while at the same time minimising the cost of introducing an edge.

In the following, we estimate the cost of adding an edge as the inverse of the overlap between the used vocabularies. This estimates the chances of finding pairs of concepts or resources based on the shared usage of predicates by the respective nodes. Intuitively, this cost measure favours "meaningful" edges, i.e. edges between nodes with overlapping vocabularies. Of course, this is a very rough estimation, that could be changed for a more accurate one without impairing the applicability of our algorithms.

Definition 6 (vocabulary of a node). *The vocabulary of a node n from either a hostnames graph $\mathcal{H}$ or a namespaces graph $\mathcal{S}$ is the set of predicates used to describes the resources contained in the node.*

$$vocab(n) = \{p \mid \exists \langle r, p, o \rangle, r \in cont(n)\}$$

Our semantic cost for a link between two nodes will be based on the similarity of the vocabularies used in the nodes. We used the standard Jaccard measure to quantify the similarity between vocabularies. This is a measure commonly used in the ontology mapping domain.

Definition 7 (Vocabulary Similarity). *The similarity $S(n_1, n_2)$ between two nodes n_1 and n_2 from either the hostname graph or the namespace graph is defined as:*

$$S(n_1, n_2) = \frac{|vocab(n_1) \cap vocab(n_2)|}{|vocab(n_1) \cup vocab(n_2)|}$$

The corresponding cost of the edge, $\langle n_1, n_2 \rangle$, is defined as the complementary of the similarity between the nodes:

$$cost(\langle n_1, n_2 \rangle) = 1 - S(n_1, n_2)$$

Of course, we could use any other measure for semantic overlap from work in ontology alignment [6], and again these could be easily plugged into the algorithms we will describe next.

Using these calculations as our basis we now define the optimisation problem as follows:

$$\text{minimize } B(< N, E' >) \text{ subject to } min \sum_{e \in E'} cost(e), \text{ where } E' = E \cup (N \times N)$$

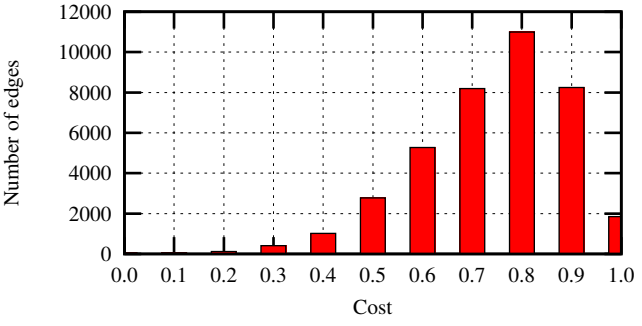
Note, that E' is the union of the existing edges with some set of newly introduced edges from the space of all possible edges in the graph.

4.2 Strategies for Adding New Edges

In order to put this strengthening of the WoD into a reasonable setting, the recommended fixes proposed hereby give an answer to the following question:

Cost	Count
0.0	54
0.1	56
0.2	126
0.3	408
0.4	1018
0.5	2780
0.6	5276
0.7	8194
0.8	10994
0.9	8246
1.0	1854

(a) numerical



(b) graphic

Fig. 2. Distribution of the costs for the edges than can be added to the namespaces graph

If I have computing resource available to create X new edges, what is the best way to spend them? In the following sections, we highlight some of the strategies that can be considered to give an answer. All the strategies have in common that they work on a finite set of edges that can be added. A graph made of n nodes can only have, at most, $n * (n - 1)$ directed edges if loops are avoided. When computing the list of edges to add, those already existing are not considered.

Note that, in this paper, we focus on the identification of links rather than their publication. The publication of links could be done by the data publishers or by a third party service (as done by Jaffri, Glaser and Millard in [11]).

Greedy Strategies

Start with the cheapest. This first strategy consists in sorting all the edges by their increasing cost and adding them one by one, stopping after X edges have been added. The rationale is that focusing on the cheapest connections will get the best reward/cost ratio for spending the resources available. This nodes are estimated to share the same vocabulary to describe their resources, linking them should increase the density of the clusters they are part of.

The implementation of this strategy requires the enumeration of all of the possible edges and sorting of them according to their cost. We implemented this as a greedy algorithm that computes the cost of all new edges, sorting them and inserting them one by one, measuring the centrality gain after each insertion.

Start with the most expensive. This second strategy is the exact inverse of the previous one. Instead of adding the new edges by increasing cost, the most expensive are added first. Linking nodes which are dissimilar should create bridges among different clusters, thereby diminishing the importance of the existing hubs already connecting these islands. The algorithm implementing this strategy is similar to the previous one and has the same scalability constraint.

Selective Strategies

Choose randomly. Rather than focusing on the cheapest or the most expensive nodes, it could be interesting to select a sample of X of them with different costs. The expected result is to mix bridging some clusters and increasing the density of others. The easiest most straightforward approach is then to randomly select the set of edges to create.

The algorithm implementing this strategy simply creates a set of new edges by sampling two random values between 1 and n. If the drawn edge is already present in the graph or in the set of edges to add, the process is repeated.

Choose wisely. This last strategy accounts for a property ignored by all other strategies: the fact that some edges could be nice to add *in combination with* others. Indeed, the centrality gain is likely not to depend only on how many new edges are created but also on *which ones*. The idea then is not to only select the edges to add one by one but to focus on a group of edges of size X, all at once.

Instead of creating only one set of edges like in the random selection, several sets are investigated in parallel and iteratively improved. This search strategy is done by an evolutionary algorithm, a population based class of algorithm known to perform well on combinatorial optimisation problems [5]. The outline of the evolutionary algorithm, a standard one, is detailed in Algorithm 1. It is a generational evolution with an elitism of 1: every new generation replaces the previous set of candidates with the exception of the best one which is kept.

4.3 Repair of the Namespaces Network

The namespaces network contains 198 nodes for 936 edges, leaving room for $198 * 197 - 936 = 38070$ new edges. The Figures 3 and 4 reports the result of the previously introduced strategies on that network.

The two greedy strategies are compared in figure 3. It can be observed that none of these baselines perform very well in two aspects: (1) many links must be added before obtaining a reasonable improvement of the centrality. 2500 links have to be added to halve the centrality. (2) both strategies first create more damage than improvements. The centrality first increases before going down again. Also, this behaviour is monotonic only after a minimum of edges have been added meaning that these strategies are only applicable if a minimum amount of resources are available. There is however a clear winner on this picture: adding edges by increasing cost is the best approach, damaging less of the network and decreasing its centrality starting at 125 edges. It can thus be concluded that focusing on the easiest pairs is best idea when one can not do better and X is large enough.

Choosing which edges to add is one way to do better than the greedy strategies. The results from the two selective strategies are reported in Figure 4. Our first observation is that both strategies outperform the greedy approaches: they are less damaging and reduce centrality faster. The random choice technique has some uncertain behaviour when less than 250 edges are added but is guaranteed

Algorithm 1. Main loop of evolutionary search strategy. The $\oplus$ is a "one-point crossover" operation than mixes two candidate solutions.

```
Initialise population P;
while not terminated do
    /* Evaluation of current sets                                   */
    foreach Candidate set of edges s in P do
        compute C_B(<N,E∪s>) / C_B(<N,E>)
    /* Creation of new sets                                         */
    P' ← best individual from P;
    while Size of P' different than size of P do
        switch with a probability of 0.1 do
            s ← tournament selection from P;
            s' ← tournament selection from P;
            P' ← P' ∪ s ⊕ s''
        switch with a probability of 0.8 do
            s ← tournament selection from P;
            foreach edge s_i of s do
                switch with a probability of 0.1 do
                    s_i ← randomly created new edge
            P' ← P' ∪ s;
    /* Generation replacement                                       */
    P ← P';
```

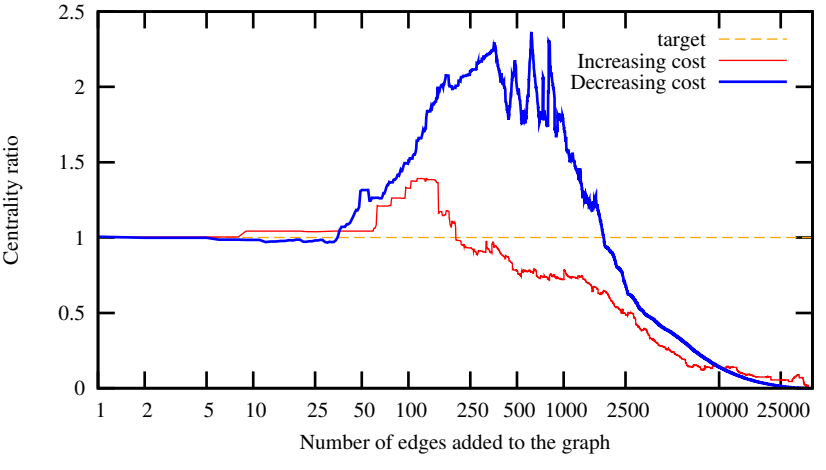


Fig. 3. Comparison of the two greedy strategies that consist in sorting all the edges according to their cost and insert them one by one, by (in/de)creasing cost

to decrease the centrality by almost 60% if at least 1000 edges are created (*e.g.* 2% of the amount of possible new edges). Both algorithms monotonically improve the centrality as soon as more than 250 edges are added. That is around 30% of the existing 936 edges. Above 10000 new edges, there is no difference in the results. For less than 250 new edges, the evolutionary algorithm finds the best sets. It achieves the best performance, decreasing the centrality by almost 60%, with a set of only 64 edges.

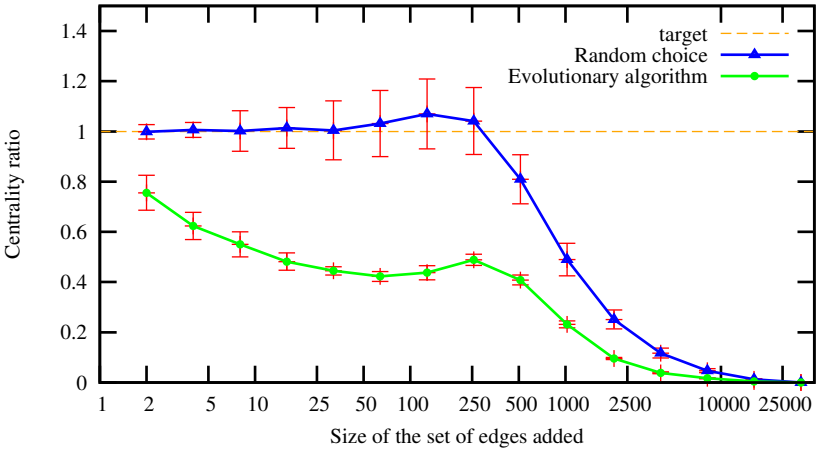


Fig. 4. Comparison of the two selective strategies applied to the namespaces network. They consist in creating a set of edges to add, either by random choice or iterative construction (evolutionary algorithm). The goal is to bring the ratio, at least, below 1.0 and, at best, close to 0.

Table 4 shows the four links recommended to create in order to decrease the centrality of the network by 30%. We now discuss whether the addition of the suggested links is feasible. Row ❶ suggests creating a link from the Lifecycle Schema to Freebase. The Lifecycle Schema describes the specification of a generic lifecycle for a resource. It defines notions such state, transition and task. Links could easily be created from this schema to descriptions of the corresponding concepts in Freebase. For example, one could link to the definition of Finite-state machine in Freebase (i.e. `http://rdf.freebase.com/ns/finite_state_machine`). Row ❷ recommends creating a link between annotations about papers from ISWC 2004 to the Ubiquitous Applications Location Ontology. This seems reasonable since one could describe the papers as having been presented at a particular geolocation, which this location ontology supports. An important note is that the given link for ISWC 2004 annotations is no longer operative. It should probably be updated to the Semantic Web Dogfood site. This is another example where old links cause the Web of Data to break. The third recommendation, Row ❸, suggests adding a link between a site describing labels for about 1 million commodities to SKOS-XL (an ontology for describing labels). A connection

between these sites again seems reasonable as one could possibly describe these commodity labels as subclasses of skosxl:Label. Finally, the recommendation, Row ❹, to link the Dublin Core types to the Cyc Ontology also could be done given that the Dublin Core types describe generic types such as Event, Image, Sound, which also appear in Cyc.

Table 4. When added all together to the namespaces graph, these 4 edge brings the centrality to 70% of its original value

	From namespace	To namespace	Cost
❶	`http://purl.org/vocab/lifecycle/schema#`	`http://rdf.freebase.com/ns/`	0.999803
❷	`http://annotation.semanticweb.org/2004/iswc#`	`http://www.w3.org/2007/uwa/context/location.owl#`	0.892857
❸	`http://openean.kaufkauf.net/id/`	`http://www.w3.org/2008/05/skos-xl#`	1.0
❹	`http://purl.org/dc/dcmitype/`	`http://sw.opencyc.org/concept/`	1.0

4.4 Repair of the Hostnames Network

The hostnames network contains 558784 nodes for 656012 edges, leaving room for $558784 * 558784 - 656012 = 312238902644$, 312 Billions, new edges. Unfortunately, such a huge number of edges makes search by enumeration impossible and the greedy approaches inapplicable. Instead, we only apply the selective strategies.

For the random strategy, as long as the number of edges added reaches 100M (that is, 0.03% of the 312B possibilities), it does not matter which ones are added. In every case, the centrality is diminished by at least 90%, going to 10% of the original value. This applies similarly for the evolutionary strategy, however, that strategy performs slightly better than the random strategy. Unfortunately, both strategies have a significant adverse impact on the hostname network before any improvement is seen for less than 100M edges added and no impact for less than 10k edges.

5 Conclusion

We can divide the conclusions of this paper into two categories: (i) *generic methods* for analysing the Web of Data, and (ii) *specific observations* on the state of the current Web of Data.

Generic methods for analysing the Web of Data

- We have defined two useful abstractions over the Web of Data, the hostname graph and the namespace graph, allowing us to analyse both the infrastructural and its semantic connectivity of the Web of Data.

- Following insights from network analysis, we have proposed betweenness centrality as the key metric for measuring network robustness (= the ability to maintain connectivity after removal of nodes).
- We have phrased the problem of improving the robustness as an optimisation problem, aiming to minimise the graph's centrality index under minimal cost of adding links. We proposed as a cost-function the Jaccard distance measure based on vocabulary overlap, but our approach is neutral as to the choice of the cost-function.
- We investigated the feasibility of a number of algorithms to solve this optimisation problem, and showed that, in particular, the use of an evolutionary algorithm was successful in identifying a small number of links that substantially increase the robustness of the graph.

Observations on the state of the current Web of Data. Assuming that the BTC dataset is indeed a representative snapshot, the following facts have been revealed by our analysis:

- The vast majority of triples on the Web of Data do not contribute to it being a web, but instead point to literals or blank nodes, or refer only to URI's internal to the same dataset. This concerns as much as 80% of all triples.
- The Web of Data is currently not a scale-free network. It shows a more extreme distribution, although it has some of the typical properties of a scale free network, in particular the presence of hub-nodes.
- Almost all infrastructural connectivity on the WoD is mediated by 3 servers, `xmlns.com`, `dbpedia.org` and `purl.org`, making the system very brittle.
- Similarly, almost all semantic connectivity is provided through a small number of namespaces, again a very brittle structure.
- On the positive side, the robustness of the Web of Data can be improved drastically: the centrality of the namespace graph can be improved by a factor of 2 by adding just 4 edges to the namespace graph.
- For the hostnames graph, we were not able to find any such easy fixes. In fact, it seems that the hostnames graph will need substantial (and hence automated) extension for it to become more robust.

Future Work. A first task would of course be to extend this work to larger snapshots of the Web of Data, to see if our methods scale and if our findings generalise. Currently, the hostname graph is already at the limits of what is computationally feasible to solve the link-optimisation problem. In particular, repeatedly testing the centrality index of candidate graphs that are generated by our evolutionary algorithm is very expensive. An incremental algorithm calculating the centrality index of a slightly modified graph would be helpful here.

A more fundamental extension to our work would be to change our analysis into a real-time monitoring engine that would constantly monitor the state of the Web of Data, *e.g.* by taking as input a stream of modifications, and produce as output a set of suggestions for useful links to add in order to maintain or improve

robustness. Unlike the regular Web, where failure is tolerated, the Web of Data is meant for machine consumption, implying that it is more in need of constant and machine-assisted upkeep. In this paper, we have provided the necessary abstractions for such quality control, and we have shown that the Web of Data in its current form has severe vulnerabilities. We have also proposed effective algorithms for determining repairs. With these results our paper opens the way towards continuous and machine-assisted repairs to the Web of Data.

In some cases adding a link may be less expensive than deploying a mirror. While studying the cost of adding links versus that of deploying mirrors goes beyond the scope of this work, we plan to work on the automated identification and connection to cached data.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature 406(6794), 378–382 (2000)
2. Amaral, L.a., Scala, A., Barthelemy, M., Stanley, H.E.: Classes of small-world networks. Proceedings of the National Academy of Sciences of the USA 97(21), 11149–11152 (2000)
3. Bader, D., Kintali, S., Madduri, K., Mihail, M.: Approximating betweenness centrality. In: Bonato, A., Chung, F.R.K. (eds.) WAW 2007. LNCS, vol. 4863, pp. 124–137. Springer, Heidelberg (2007)
4. Bader, D., Madduri, K.: SNAP, Small-world Network Analysis and Partitioning: an open-source parallel graph framework for the exploration of large-scale networks. In: IEEE International Symposium on Parallel and, pp. 1–12. IEEE, Los Alamitos (April 2008)
5. Eiben, A., Smith, J.: Introduction to evolutionary computing. Springer, Heidelberg (2003)
6. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
7. Freeman, L.C.: A Set of Measures of Centrality Based on Betweenness. Sociometry 40(1), 35 (1977)
8. Ge, W., Chen, J., Qu, Y.: Object Link Structure in the Semantic Web. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 257–271. Springer, Heidelberg (2010)
9. Gil, R., Garcia, R.: Measuring the semantic web. In: Advances in Metadata Research, Proceedings of MTSR 2005. Rinton Press (2006) ISBN 1-58949-053-3
10. Guéret, C., Wang, S., Schlobach, S.: The web of data is a complex system - first insight into its multi-scale network properties. In: Proceedings of the European Conference on Complex Systems, ECCS (2010) (to appear)
11. Jaffri, A., Glaser, H., Millard, I.: Uri identity management for semantic web data integration and linkage. In: 3rd International Workshop On Scalable Semantic Web Knowledge Base Systems. Springer, Heidelberg (2007)
12. Newman, M.E.J.: The Structure and Function of Complex Networks. SIAM Review 45(2), 167–256 (2003)
13. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 707–716. ACM, New York (2007)