A Spectrometry of Linked Data

Giovanni Bartolomeo
University of Rome Tor Vergata
Via del Politecnico, 1
00133 Rome, Italy
giovanni.bartolomeo@uniroma2.it

Stefano Salsano
University of Rome Tor Vergata
Via del Politecnico, 1
00133 Rome, Italy
stefano.salsano@uniroma2.it

ABSTRACT

Entity mining is still a troublesome open problem. In past years many approaches allowed to automate the generation of equivalence links between references using schema matching or various heuristics based on the recognition of similar property values. In contrast, few of them considered the analysis of the network of equivalence links ("equivalence network") as an indication of the likelihood and strength of the equivalence. Following this basic idea, in this paper we apply the well known Girvan and Newman algorithm to partition existing equivalence networks into clusters of co-references and gain an insight of their nature, size and composition.

Categories and Subject Descriptors

H [**Information Systems**]: Models and Principles; H.1 [**Models and Principles**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval — information filtering.

Keywords

Entity, Co-references, Linked Data

1. INTRODUCTION

Could a URI reference (URIRef) be thought as exactly "attached" to its referent? Could it make sense to talk about entity "identifiers" or would it be better to talk about more ambiguous "references", i.e., placeholders for any model that satisfies the formal semantics of the Semantic Web (Hayes)[1]? Booth [1] observes that the aforementioned question, which in the past has been often regarded as fundamental in the debate about identity on the Web, is relatively unimportant. As long as an entity, identified by whatsoever URIRef, is associated to at least one description containing machine understandable information, this information can be automatically processed and used by applications.

Yet the proliferation of references poses a practical problem in Linked Data. From [2] we learn that using multiple references for the same entity (in short, "co-references") is a fault-tolerant approach, lowers the barrier to enter the Linked Data and helps in maintaining traceability of different "views" of the same entity by various data publishers. On the contrary, the opposite party [3] observes that co-references make difficult entity "consolidation". In fact, allowing anyone to issue a new URIRef for any entity results in the open problem of objectively stating the degree of matching of different descriptions about the same entity [4].

In the past the vision of a single, canonical "entity identifier" has inspired at least two major European projects (the ReSIST Network of Excellence[2] and the OKKAM project[3]). In [5], for instance, Jaffri describes a "Consistent Reference Service" (CRS) that aggregates entity co-references into bundles. Each bundle contains only one preferred reference ("canon") and a number of other co-references ("duplicates"). To obtain "consistent references" at a scale, the CRS recommends using the canon instead of duplicates; in many cases, however, the canon is a random choice of the system and is no more "representative" than its duplicates. Differently, Bouquet [6] proposes a solution based on "OkkamIDs", a new class of identifiers that "directly refer" to entities. As opposite to different descriptions provided by single data publishers, the notion of direct reference is realized by means of a shared entity profile, i.e. an associated description, accessible by dereferencing the OkkamID and containing information agreed and consolidated by the Web community. Being shared by the community of its users, the entity profile tends to become as much complete and exhaustive as possible. According to Bouquet, this feature would provide the answer to the argument, raised in [7], that the user (i.e., the data publisher or consumer) tends "*to observe* [only] *a small portion of* [a reference] *use*"; thus, implicitly, her knowledge about the referenced entity remains ambiguous.

At the time of writing, however, neither the CRS nor OkkamIDs seem to have provided the ultimate solution for entity identification. Nevertheless, they contributed to raise an interesting multidisciplinary discussion on this topic and to provide good theoretical and practical inputs to several related researches. In particular, the main lesson learned from these experiences was that community (i.e. inter-domain) consensus is fundamental. This realization seems to suggest that the task of evaluating an equivalence link as correct or incorrect often requires a global perspective. A "bird-eye view" of the entire set of co-references and of their equivalence links could probably provide more useful indications about the strength of a given equivalence. As a matter of fact, the Linked Data cloud has begun

[1] See for example P. Hayes. Message to www-rdf-comments@w3.org,2003. http://lists.w3.org/Archives/Public/www-tag/2003Jul/0198.html (accessed March, 21 2011).

[2] The ReSIST European Newtork of Excellence co-funded by the European Commission (GA 026764) ran from January 2006 to March 2009, http://www.resist-noe.org/ (accessed March, 21 2011).

[3] The OKKAM project co-funded by the European Commission (GA 215032) ran from January 2008 to June 2010, http://www.okkam.org/ (accessed March, 21 2011).

LDOW2012, April 16, 2012, Lyon, France.

to show widely referred nodes tending to attract a large portion of incoming equivalence links, and becoming, for the Linked Data community, more representative of an entity than others. At a closer look, however, this observable fact presents at least three particular features: it is distributed, because, for each entity, more than one node might aspire to become an "authority". It is also dynamic, as it varies, over the time, according to the raising of new pay-level domains[4] publishing their entity descriptions as Linked Data. Last but not least, it is aggregative: groups of co-references, coming from different domains, tend to aggregate into clusters. But, despite its popularity, few works have focused on the analysis of this phenomenon. Actually, previous works, which we will describe in section 2, have provided similar investigations on equivalence links; but none of them, to the best of our knowledge, analyzed the tendency, shown by co-references, to form clusters. Therefore, the main contribution of this paper will consist in getting a first insight into this trend – ignoring, for the time being, its dynamics. To this end, in section 3 we will illustrate a methodology to detect clusters of nodes in equivalence networks (i.e. networks formed by equivalence links) based on the well known community detection algorithm developed by Girvan and Newman. Next, in section 4, we will present the results of an experiment made applying the algorithm to a dataset of about 1.7 million equivalence links that we collected by means of the Sindice[5] search engine API. Then we will discuss some interesting observed patterns, and propose a synoptic diagram showing the average composition of a cluster as a function of its cardinality (section 5). Finally we will draw the conclusions and hint at future developments and possible applications of our methodology.

2. RELATED WORKS

Entity matching has been addressed mainly by providing tools to automate the generation of equivalence links, using schema analysis [8] or heuristics that recognize similar property values [9]. These works are indeed related to our approach: by publishing equivalence links on the Linked Data in form of `owl:sameAs` statements, they served to progressively build the precious ground relevant for the cluster analysis we are going to present.

However, contrary to the synthesis of new potential equivalence links, we are interested in a methodology that allows isolating groups of co-references perceived by the Liked Data community as "consistent", i.e. as mostly referring to given entity.

Analysis of existing similarity and equivalence links has been performed in the past by Hu (2008) [10] and Ding (2010) [11]. In particular, Hu considered not only explicit equivalences (i.e., RDF statements containing the `owl:sameAs` predicate), but also other predicates that may provide hints on the equivalence of two resources, namely inverse functional properties, functional properties and maximum cardinality. However, his experimental results on a large-scale dataset (76 million URIrefs) have shown that the bulk (99.8%) of equivalence relations is given by explicit statements. Interesting enough, Hu also considered the "authoritativeness" of the context in which the equivalence statements appeared, showing that only 6% of the over 7 million equivalence statements he analyzed appeared in RDF documents reachable by dereferencing the subject as well as the object of the statement, 66% by dereferencing either the subject or the object whereas 27% were in other documents. Therefore, an indexing service (such as e.g., Sindice) is generally needed in order to discover existing statements from these documents as well.

Ding was the first researcher that used the term "sameAs networks" to denote those RDF graphs formed by only RDF statements containing the `owl:sameAs` predicates. He performed a statistical investigation explicitly focused on these networks involving about 8 million equivalence links among nodes arranged into 2 million weakly connected components. He found that the in-degree (i.e. the number of incoming `owl:sameAs` links per node) distribution exhibited the power law pattern characteristic of scale-free networks. Another result in Ding's experiment was that the highly referenced nodes were from relatively few domains such as dbpedia.org, freebase.com, geonames.org.

3. PROPOSED APPROACH

We need to look at the development of node "clusters", i.e. groups of nodes within which equivalence links are much more dense than between them. A similar phenomenon has been found in many complex networks [12], finally encouraging the two physicians Newman and Girvan [13] to develop an ad hoc algorithm facilitating cluster detection. Before introducing their algorithm, however, we need to formalize the concept of graph, path, connected component, modularity and edge betweenness.

Let I, B, L be disjoint infinite sets of URIRefs, blank nodes and literals.

*Definition 1. **RDF Graph**.* A RDF triple is a tuple (s,p,o) ∈ I ∪ B x I x I ∪ B ∪ L, with s ∈ I ∪ B, p ∈ I, o ∈ I ∪ B ∪ L, and s said subject, p said predicate, o said object. An RDF graph is a set of RDF triples. A subject or an object of a RDF triple is called a node of the graph.

*Definition 2. **RDF Subgraph**.* A subgraph of a RDF graph G is a RDF graph whose RDF triples are a subset of those in G.

*Definition 3. **SameAs Network**.* A sameAs network is a RDF graph whose RDF triples are all in the form (s,`owl:sameAs`,o).

*Definition 4. **Arcs and Edges**.* Let G be an RDF graph. A predicate in a RDF triple in G is called an arc of G and is represented as a direct link from the subject to the object. An edge of G is any undirected link between two nodes.

In the following of our discussion it will be not critical to consider the direction of the links in a RDF graph G. Therefore we will assume that for each arc in G connecting two nodes m and n there is always an associated edge that connects m with n.

[4] Pay-level domain is the term used to identify a domain subordinate to a generic top level domain or to a country code top level domain. In the remainder of this paper, and for the sake of simplicity, we will often use the term "domain" to refer to a pay-level domain.

[5] Sindice, The Semantic Web Index, http://sindice.com/ (accessed March, 21 2011).

Definition 5. ***Neighbours***. Two nodes that are connected by an edge are said to be neighbours.

Definition 6. ***Undirected Path***. An undirected path is a sequence of edges that connects one node to another.

Definition 7. ***Connected Component***. A (weakly) connected RDF graph is a RDF graph where there exists an undirected path between any pair of nodes. A (weakly) connected component of a RDF graph G is any RDF subgraph of G that is connected.

Definition 8. ***Partitions and Partition Set of a Graph***. A partition set of a graph G is any subgraph G' obtained from G by removing as many arcs as needed to fragment G into a set of exhaustive disjoint connected components, said partitions, S_i, so that $1<i<|G|$, $G'=\bigcup S_i$, $S_i \cap S_j = \phi$ $i \neq j$.

In practice, a partition set represents one of the possible dissections of the original graph into sets of nodes called partitions. Now, our problem is to find the "best" partition set, i.e. the one which most closely depicts the tendency of the nodes in the original graph to arrange themselves into clusters. To this end, let us consider a RDF graph G of cardinality |G| and let G' be one of its partition sets. Let $e=\{e_{ij}\}$ be the symmetric matrix whose element $e_{ij} \in [0,1]$ represents the fraction of total edges, in G, connecting nodes that in the partition set G' would belong to partition S_i with nodes that in G' would belong to S_j. The element e_{ii} represents the fraction of edges connecting nodes within partition S_i. Denoting with Tr(x) the trace of the matrix x, $\sum(x)$ the sum of its elements and $x \cdot y$ the multiplication of matrices x and y, we can enunciate the following definition and theorem:

Definition 9. ***(Girvan and Newman) Modularity***. The quantity $Q = Tr(e) - \sum(e \cdot e) \in [0,1]$ is called modularity.

Theorem 1. Given a partition set G' of a graph G, the modularity in Definition 9 quantifies the fraction of edges in G falling inside the partitions of the partition set G' minus the expected value that the same quantity would have in a graph H having the same partition set G' but random connections between all its nodes.

Demonstration of this theorem is given in [13]. Intuitively, the modularity estimates "how much" the arrangement of links falling in the considered partitions differs from a random pattern. Therefore, to detect a meaningful "community structure", i.e. a tendency of nodes to aggregate into groups – instead of falling randomly – Q should present relatively high values, say in the range [0.3,0.8]; According to Newman and Girvan, in fact, values greater than Q=0.8 have not yet been found in any natural network.

We now finally define the notion of cluster.

Definition 10. ***Cluster***. A cluster is any partition S_i in the partition set G' of a graph G that shows the optimal (highest) modularity.

Clusters are therefore the partitions of the partition set(s)[6] that maximizes the modularity. Note that the modularity is a property defined for the graph itself, with respect to the considered partition set. It is not a property of each single partition in the partition set.

In order to detect clusters, Newman and Girvan proposed an algorithm based on the removal of edge presenting the highest "betweenness", which is below defined.

Definition 11. ***Edge Betweenness***. The betweenness of edge e is the number of shortest paths between pairs of nodes that run along it.

Note that edges with high betweenness are likely to connect different clusters, because they are part of the maximum number of shortest paths connecting nodes from different clusters.

The simplest algorithm proposed by Girvan and Newman progressively computes the edge e with highest betweenness in a graph G, removes it and computes the modularity of the resulting graph G' (a partition set of the original graph) until the highest modularity value is found. In table 1 we propose a slightly modified version of this algorithm which returns the maximum modularity value in partition sets with less than n_{max} connected components. This algorithm considers only partition sets that exhibit modularity greater than a given threshold Q_{min}. To collect more meaningful results, we decided to feed the algorithm with only sameAs networks with a sufficient number of edges (safely we considered only networks with cardinality greater than 15 nodes). Furthermore we experimentally set n_{max}=6 (likely a reasonable value compared to the original semantics of `owl:sameAs`) and Q_{min}=0.35.

4. EXPERIMENTAL RESULTS

We collected our dataset from Sindice using Sindice4J API (a Java wrapper for Sindice Search and Cache API). We performed all computations on a machine equipped with i686 Intel Xeon CPU 3060 2.40GHz processor, 1,048,772kB RAM, and featured Gentoo Linux Base System 1.12 OS, kernel 2.6.18-xen, Java v6.0 and Aduna Sesame v2.60 back end by Gentoo Linux MySQL v5.1. Graph algorithms were implemented using JUNG API v2.0. The hardware equipment we used was definitely cheaper than the one described in previous experiments, thanks to the choice of using Sindice Cache API and storing locally only equivalence statements instead of caching full datasets.

In Table 2 we provide some statistics of our experiments compared with the ones presented by Hu [10] and Ding [11]. The total amount of RDF statements, calculated by using the methods provided by Sindice Cache API, was 314,922,780 – certainly less than the one handled by Ding, but comparable with the number of statements considered by Hu. The number of different `owl:sameAs` statements found in the dataset was 1,722,938, representing 0.55% of the total amount of statements, a percentage closer to the one reported by Ding than the one provided by Hu.

[6] There might be more than one partition set showing the same modularity.

Applying the algorithm in Table 1, we restricted the dataset to only sameAs networks with cardinality greater than 15 nodes and having a modularity Q>0.35 when split in 6 clusters at most. In our dataset 2,922 networks presented these features.

Table 1. A cluster detection algorithm. The algorithm iteratively removes the edge with the highest betweenness and computes the modularity of the resulting graph until the highest value is found.

```
Input: G, a sameAs network
Output: a set of clusters in G

$n_{max}$ ← 6
$Q_{min}$ ← 0,35
Q ← 0
if |G|<16 then return void
repeat {
      e ← edge with highest betweenness in G
      remove e from G
      if modularity(G)>Q then Q ← modularity(G)
} until (number of connected components in G)<$n_{max}$
if Q>$Q_{min}$ then return (connected components in G)
else return void
```

Table 2. A **comparison of statistics of our experiment with the previous ones performed by Hu (2008) and Ding (2010). For each experiment, we report the number of equivalence statements as an absolute value and as a fraction of the total amount of considered statements.**

	Our	Ding [11]	Hu [10]
Year	2012	2010	2008
Source	Sindice	BTC 2010	Falcons
RDF statements	314,922,780	3,171,184,130	596,418,935
Eq. statements	1,722,938	9,358,227	7,880,906
Ratio	0.55%	0.30%	1.32%

About 80% of the networks were fragmented in either five or six clusters; the remaining ones were partitioned into three or four clusters; none into two clusters. The maximum modularity ranged from 0.35 to 0.67; its distribution, shown in Figure 1, presented two peaks around 0.51 and 0.62 (respectively 24% and 19% of the whole population).

Proceeding with a manual inspection we found that the algorithm isolated clusters more precisely referring to an entity from clusters containing less specific references. For instance, the conceptual entity “sergeant” was found in a sameAs network partitioned into four clusters with modularity Q=0,44. The biggest cluster was mainly formed by nodes from dbpedia.org, suggesting slightly different meanings for the same concept: sergeant instructor, detective sergeant, senior sergeant, etc. The second relevant cluster, more consistently referring to the concept of police sergeant, was formed by six co-references from five different domains.

Sometimes our detections revealed different location entities wrongly stated to represent the same place. For instance, the component referring to Abuja, since 1991 the new capital of Nigeria, mistakenly appeared in the sameAs network containing Lagos, the former capital of the country. The algorithm clearly isolated Abuja from other two clusters referring to Lagos with a maximum modularity Q=0.49.

Similarly, the network containing the node `<http://umbel.org/umbel/sc/Italy>` was partitioned into three clusters with maximum modularity Q=0,49. One of the clusters however contained only co-references to Mendocino, a Victorian village near San Francisco, CA, renowned for its award winning wines.

5. DISCUSSION

After the manual inspection we noted that edges with high betweenness generally corresponded to mistakenly added `owl:sameAs` links. However, sometimes removed links were from the collections maintained in okkam.org, the entity search portal created by the OKKAM project. Nodes from this domain were often acting as "hubs", showing an elevated number of outgoing arcs but no incoming links. They frequently appeared in paths connecting two or more clusters. Thus, their outgoing arcs, presenting an elevated betweenness, were often removed.

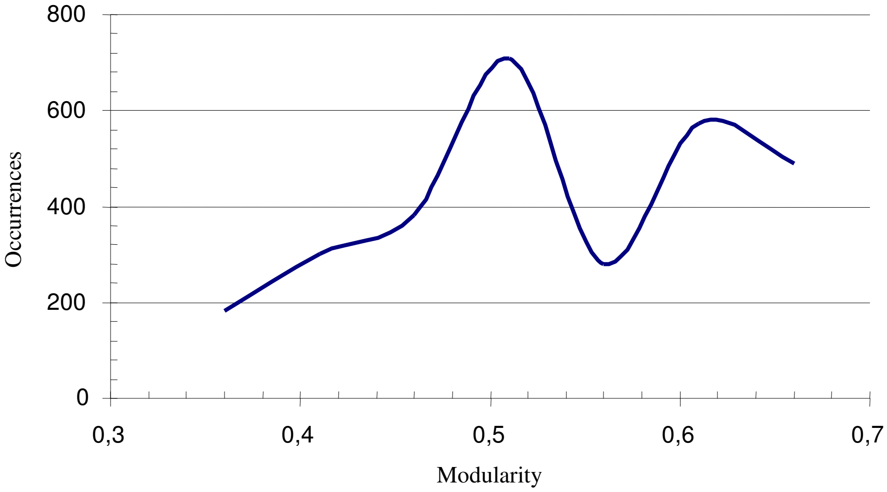

Figure 1. Distribution of the highest modularity in about three thousand sameAs networks with cardinality greater than 15 nodes and modularity Q>0.35 when split in 6 clusters at most.

Almost all the networks presented two typical clusters: The first one characteristically contained nodes from dbpedia.org and fu-berlin.de that were often linked to a central node from freebase.com (or sometimes from mpii.de). The second typical cluster included a smaller number (about 5-10) of heterogeneous nodes coming from various domains (umbel.org, opencyc.org, nyt.com, freebase.com, mpii.de, etc.)

To account for this phenomenon, we measured out the average composition of each cluster and plotted it as a function of the cluster's cardinality. The result was the "cluster spectrometry" depicted in Figure 2.

This diagram shows three main different areas. Clusters with cardinality 1 or 2 are prevalently made of nodes coming from dbpedia.org and fu-berlin.de. In particular, clusters with cardinality 1 account for isolated nodes, whereas those with cardinality 2 quite often represent couple of nodes interlinking resources in these two domains.

The contribution from more domains becomes evident from clusters with cardinality greater than 2, which begin to show a more heterogeneous composition. From cardinality 3 to about cardinality 12, dividing the cluster's cardinality by the number of involved domains we obtain a ratio close to 1, which means one node per domain on average. We observe that these clusters likely represent the "core" of Linked Data, as they contain interconnected nodes from several different domains describing the same entity. These clusters prove the existence of a set of well aligned "synonyms", which, together with their associated descriptions, contribute to determine the generic meaning of an entity as shared by the Linked Data community.

Clusters with cardinality greater than 12 progressively drop this characteristic composition and begin to be more homogeneous. For instance, we can observe how contributions from freebase.com, nyt.com and opencyc.org progressively decrease with the number of nodes, in favor of clusters generally dominated by nodes from dbpedia.org and fu-berlin.de. Moreover, these clusters often consist of hyponyms (i.e. more specialized terms) connected to a central node but not interlinked each others. One trivial explanation for this is that, at the time of writing, only DBpedia provides finer granule definitions while other data providers often publish more generic definitions. However, these semantic nuances have not been captured during the linkage and have been flattened as "equivalences" with the corresponding hyperonym (i.e. more generic tem).

We are aware that our investigation covered only 1% of the over 30 billion RDF statements in Linked Data and possibly even fewer samples having restricted the study to only networks compliant with the criteria illustrated in section 3. However, we believe that the discovered behavior could be also found in larger portions of Linked Data and we plan to perform a more exhaustive analysis extending our investigation to larger datasets. Possible criticism is also related to the bias introduced by Sindice. Many RDF statements that have been cached by Sindice are no more in the Linked Data cloud, which is continuously refining and evolving. Many faulty equivalences seem to have been corrected on the dereferenceable online version of the source documents. Clearly this bias can be reduced by repeating the experiment and using a more recent version of Sindice caches. Nevertheless, we noted that the presence of faulty equivalences represented a good benchmark for our algorithm, which was able to clearly detect clusters consistently referring to an entity and to keep them separate from others.

6. CONCLUSIONS AND FUTURE WORKS

Entity consolidation has been subject of many researches in past years, with many efforts focused on providing algorithms and tools to automate the generation of equivalence links in Linked

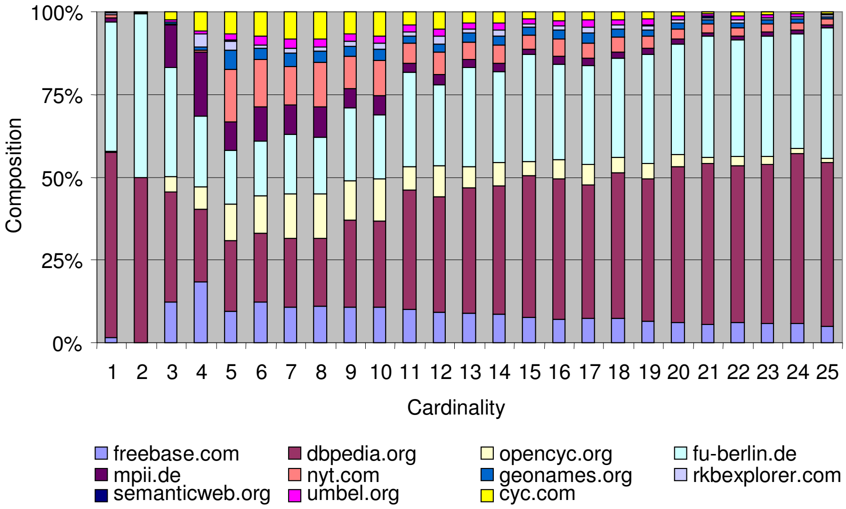

Figure 2. Average composition of clusters as a function of their size. The clusters prove the existence of a set of well aligned synonyms, which, together with their associated descriptions, contribute to determine the meaning of an entity as shared by the Linked Data community.

Data. Now that Linked Data has passed its bootstrapping phase, with several billions of equivalence links available, link analysis could provide novel meaningful results to previously unanswered questions.

In this paper we used the already deployed equivalence links to run a cluster detection algorithm based on edge betweenness and Newman and Girvan modularity.

Analyzing the results we found typical recurring clusters consisting in a small number of heterogeneous nodes that we believe to represent the bulk of consolidated entity references in the Linked Data cloud.

This finding inspired us to provide our own answer to the identity debate we mentioned in the introduction of this paper: rather than looking at an unique stable identifier for an entity, one should accept the existence of a dynamic set of "synonyms", arranged into clusters and contributing to create the "meaning" of that entity as understood by the Linked Data community. This does not preclude the possibility to detect most representative nodes (for instance, the ones with the highest in-degree and cluster coefficient [14], or authority rank [15]), but this is not strictly necessary and does not seem to offer any particular advantage except probably the "psychological" one of having solved the quest for an unique entity identifier. Most likely, the very problem behind entity identification is not "how many" identifiers, but "which ones" should be used to refer to an entity. In this paper we attempted to provide a first answer to this question from a novel perspective, the link analysis, which in our opinion might reveal several new understandings of Linked Data in the near future.

There are several future directions of investigation: for instance, one potential important finding that we highlighted was that mistakenly added links showed the highest edge betweenness. This suggests an interesting novel technique to detect misleading equivalences (and thus different entities) at a scale. However, in order to mechanize this procedure, an automatic verification step – which we have not yet implemented – is necessary. At a first glance, this verification necessarily requires a deeper analysis of the involved nodes, and of their properties other than `owl:sameAs` (for instance: literal properties, type, etc.). However, under the assumption that similar entities should present similar structures in their set of links, an interesting alternative could be using link analysis algorithms that measure the similarity between the link sets of the different nodes. This technique, called "blockmodelling", has been proposed in the past for the analysis of social networks [16].

As another direction of research, we plan to broaden our methodology by introducing more sophisticated cluster detection techniques. The Girvan-Newmann edge betweenness algorithm we used requires that each node must appear in one and only one cluster. This condition is probably too strong. Existing algorithms based on a combination of edge betweenness and vertex betweenness [17] remove this limitation and might produce more meaningful results.

Moreover we believe that other interesting features – e.g., relationships between clusters – might be unveiled by applying more advanced detection techniques such as, for instance, the "mesoscopic" analysis of connected components recently proposed by Arenas et al. [18].

7. ACKNOWLEDGMENTS

This work is partially sponsored by the European Commission under the framework of the 'Convergence' project (FP7-257123).

8. REFERENCES

[1] Booth, D. 2006. URIs and the myth of resource identity. In Proceedings of Identity, Reference, and the Web Workshop at the WWW Conference.

[2] Heath, T., Bizer, C. 2011. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory & Technology. Morgan Claypool.

[3] Bouquet, P., Stoermer, H., Niederee, G., Mana, A.2008. Entity Name System: The Backbone of an Open and Scalable Web of Data. In: Proceedings of the IEEE International Conference on Semantic Computing, 554-561, IEEE Computer Society.

[4] Halpin, H., Hayes, P.P., McCusker, J., Mcguinness, D., Thompson, H. 2010. When owl:sameas isn't the same: An analysis of identity in linked data. In Proceedings of the 9th International Semantic Web Conference.

[5] Jaffri, A., Glaser, H., Millard. 2008. URI disambiguation in the context of linked data. In Proceedings of the 1st International Workshop on Linked Data on the Web.

[6] Bouquet, P., Palpanas, T., Stoermer, H., Vignolo, M. 2009. A Conceptual Model for a Web-scale Entity Name System. In Proceedings of 9th the Asian Semantic Web Conference.

[7] Hayes, P., Halpin, H. 2008. In defense of ambiguity. International Journal of Semantic Web and Information Systems, 4(3).

[8] Shvaiko E. 2007. Ontology Matching. Springer-Verlag.

[9] Euzenat, J., Ferrara, A. et al. 2010. Results of the Ontology Alignment Evaluation Initiative 2010, http://disi.unitn.it/~p2p/OM-2010/oaei10_paper0.pdf (accessed March, 21 2011)

[10] Hu, W., Qu, Y. and Sun, X. 2011. Bootstrapping object coreferencing on the semantic web. Journal of Computer Science Technology, 26(4), 663–675.

[11] Ding, L., Shinavier, J., Shangguan Z., McGuinness, D. 2010. SameAs Networks and Beyond: Analyzing Deployment Status and Implications of owl:sameAs in Linked Data. Lecture Notes in Computer Science, Volume 6496/2010, 145-160.

[12] Strogatz, S. H. 2001. Exploring complex networks. Nature, 410, 268–276.

[13] Newman, M. E., Girvan, M. 2004. Finding and evaluating community structure in networks. In Physical Review E, volume 69, issue 2.

[14] Watts, D. J., Strogatz, S. H. 1998. Collective dynamics of 'small-world' networks. Nature, 393, 440–442.

[15] J. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604–632.

[16] Wasserman S., Faust, K. 1994. Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge.

[17] Pinney J. W., Westhead D. R. 2006. Betweenness-based decomposition methods for social and biological networks. Interdisciplinary Statistics and Bioinformatics, 87–90, Leeds University Press.

[18] Granell, C., Gómez, S. and Arenas, A. 2011. Mesoscopic analysis of networks: applications to exploratory analysis and data clustering. Chaos: An Interdisciplinary Journal of Nonlinear Science, 21, 016102.