On the Likelihood of an Equivalence

Giovanni Bartolomeo[1], Stefano Salsano[1] and Hugh Glaser[2]

[1] University of Rome Tor Vergata,
Via del Politecnico 1, 00133 Rome, Italy
{Giovanni.Bartolomeo, Stefano.Salsano}@uniroma2.it

[2] Seme4, Ltd.
18 Soho Square, London, W1D 3QL, UK
Hugh.Glaser@seme4.com

Authors' accepted manuscript, published in Lecture Notes in Computer Science Volume 8186, 2013, pp 2-11. The final publication is available at http://link.springer.com/chapter/10.1007%2F978-3-642-41033-8_2

Abstract. Co-references are traditionally used when integrating data from different datasets. This approach has various benefits such as fault tolerance, ease of integration and traceability of provenance; however, it often results in the problem of entity consolidation, i.e., of objectively stating whether all the co-references do really refer to the same entity; and, when this is the case, whether they all convey the same intended meaning. Relying on the sole presence of a single equivalence (owl:sameAs) statement is often problematic and sometimes may even cause serious troubles. It has been observed that to indicate the likelihood of an equivalence one could use a numerically weighted measure, but *the real hard questions of where precisely these values come from* arises. We propose and discuss an answer to this question.

Keywords: Equivalence Mining, Co-references, Linked Data.

1 Introduction

Co-references (i.e. reference to the same resource) are widely used in Linked Data to integrate data from different datasets. The simplest and usual way of representing co-references is to state the explicit equivalence between two RDF nodes connecting them through the owl:sameAs property. Hu [1] noted that despite other properties such as inverse functional properties, functional properties and maximum cardinality that may indirectly confirm the equivalence of two resources, the bulk of equivalence relationships in Linked Data are traditionally given by explicit owl:sameAs statements (henceforth also called equivalence statements or simply equivalences).

Using co-references is probably unavoidable in a distributed environment[1] and brings various benefits such as fault tolerance, ease of integration and traceability of provenance [2]. However, it often results in the open problem of objectively stating whether all the co-references do really refer to the same entity (a well known problem sometimes called object identification, entity consolidation, etc.) and, most importantly, whether all co-references convey the same intended meaning, as OWL specifications[2] would require. owl:sameAs statements do not always honor this rigorous semantics. For example, [3] reports four very different typologies of use of owl:sameAs they actual found in Linked Data (contextualization, referential opacity, similitude, and reference misplacement). While these different uses appear to be acceptable and sometimes even useful for some applications (e.g. surfing the web of Linked Data for general knowledge, such as finding that Berlin is a city in Germany, together with a number of other cities), for others, especially for those using automatic reasoning, a higher degree of precision is indeed needed [4].

Looking at existing properties in well known vocabularies conveying the meaning of equivalence and similitude, one could think of a classification of the strength of this kind of relationship. For example, [5] points out that rdf:seeAlso is much "weaker" than owl:sameAs; and [3] that the SKOS vocabulary has a number of "matching" predicates that are close in meaning to owl:sameAs without however implying full identity (skos:broadMatch, skos:narrowMatch, skos:closeMatch, etc.). However, their use may be subjective and even if *one is tempted to engage with some sort of numerically weighted uncertainty measure of identity, the real hard questions of where precisely will these real values come from* [3] arises. Our first question to answer.

2 Related Works

At the time of writing there are two main ways of creating equivalence relationships: automatic and manual. Automatically generated equivalences[3] depend on the effectiveness of various adopted algorithms in detecting similar property values presented by candidate equivalent resources (and obviously on the public availability of such property values). Many of these algorithms are domain-independent and – as opposite to humans – tend to disregard semantic nuances in favour of the sought similitude. Thus, reliable links are often established after manual inspections of candidate resources.

Manually established equivalence links depends on the publisher's knowledge about the referent, the meaning they intend to convey, their understanding of the candidate equivalent resources in other datasets and her knowledge about contextual

1 Attempts to provide single unique identifiers, such as [14, 15], at the end result in centralized systems.

2 According to OWL specifications, two RDF subjects stated to be equivalent should be identical, and thus perfectly interchangeable in all the statements they appear.

3 The Ontology Alignment Evaluation Initiative website (http://oaei.ontologymatching.org/) contains articles and reports about several methods which have been compared in various schema matching campaigns starting from 2004.

aspects (for instance, when linking data from medical domains, a publisher might be well aware that a high precision is needed).

As equivalence is a transitive and symmetric property, sets of resources that convey the same meaning, seen as nodes connected by owl:sameAs arcs, should form a complete directed graph, in which every pair of distinct nodes is connected by a pair of unique arcs, one in each direction.

"Consistent Reference Service" (CRS) is a concept that has been proposed in the past to deal with equivalences [12]. A CRS is a framework that aggregates "real" co-references into bundles. A user can look for a reference and get all the co-references belonging to the same bundle (a well-known instantiation of a CRS is "sameAs.org", which is exposed both as a web site and a web API[4]).

The genesis for CRS deployment was to cope with the questions of how to deal with nodes related by owl:sameAs predicates which should form a complete directed graph, either explicitly or by inference. In combining different graphs where equivalences have been defined or identified, a CRS takes the only safe action and asserts all the (N^2) individual owl:sameAs links connecting all the possible pairs of nodes. In practice, however, other topologies are observable. In 2010, working on equivalence links connecting RDF nodes from the BTC2010 corpus, Ding [6,7] found that their distribution exhibits the power law pattern characteristic of scale-free networks, i.e. few nodes have many incoming arcs while others have much fewer ones.

3 Method

We observe [8] that co-references coming from different datasets show a dynamic tendency to aggregate into groups of graph nodes within which edges are much more dense than between them. In literature, these groups are known as "clusters" [9]. These empirical findings led us to a second question: why does this happen?

Answering this question means analyzing equivalence network using cluster detection techniques. In general, the graph clustering problem is NP-hard, therefore no algorithm exists to solve it in polynomial time; at the time of writing only heuristics are available. Newman and Girvan [10] propose the concept of modularity as a measure of the degree of clustering of a graph. The modularity is defined as the fraction of edges falling in the resulting disjoint connected components minus the expected value that the same quantity would have if the graph had random connections between nodes. They also describe an iterative procedure to identify possible clusters based on betweenness centrality of each edge[5] (number of shortest paths passing through the edge) in the graph. At each step, the edge with the highest betweenness is removed; as edges with high betweenness are likely to be the ones connecting *different* clusters, the modularity increases until a maximum value is

4 Whereas sameAs.org maintains its own CRS, it also hosts several others maintained by different organizations.

5 Many cluster detection algorithms work on undirected graphs, thus on edges, not on arcs. To account for arcs that mutually connect two nodes, we assume that the corresponding edge connecting the two nodes has a double weight.

reached. The remaining connected components are likely to represent the sought clusters. In [8] we illustrate the results obtained applying this algorithm to a subset of equivalence relationships taken from the Linked Open Data cloud. Despite Newman and Girvan's algorithm, and other similar ones, actually being able to detect clusters, unfortunately they often tend to overestimate their number; due to the choice of edge betweenness as a criterion, some topologies (e.g. star-like structures) are not seen as clusters and are thus decomposed into single nodes. A different approach recently proposed by Noack [11] seems to overcome this behavior. In Noack's approach, a graph is seen as a metaphor of a mechanical system where each node is a particle whose "position" is determined by the "forces" acting on it. Close particles tend to repulse each other; edges, which can vary their "length", provide attractive forces that tie particles together. At the equilibrium the fraction between the average edge length and the average node distance is minimized. By making edges short and distances between not connected nodes long, this approach tends to highlight clusters.

Noack identifies a whole class of "clustering" energy models which satisfy these properties and differ only for the law used in the mathematical formulation of the attractive and the repulsive forces between particles. Two relevant models of this class are "LinLog" and "QuadLin"; both of them are below discussed.

LinLog energy model. Let N(G) be the set of nodes in the graph G, x, y two arbitrary nodes, p_x and p_y their own position in a spatial system R^n. Given two disjoint non empty sets of nodes U, V such that $U \cup V = N(G)$ we denote with E(U,V) the set of pairs of nodes connected by edges (x,y) with $x \in U$ and $y \in V$. We call the pair (U,V) a bipartition of G, and define its density as

$$d_{U,V} = |E(U,V)| / (|U| \cdot |V|)$$

where $|\cdot|$ is the cardinality of a set. The LinLog energy model is then defined by the formula

$$E = \sum_{x,y \in E(U,V)} ||p_x - p_y|| - \sum_{x \in N(G),\, y \in N(G),\, x \neq y} \ln(||p_x - p_y||)$$

where E is the energy associated to the system. In LinLog, initially, the position of each node is set using a random function; the position then evolves according to the forces acting on each node determined by the energy model. Noack demonstrates that, when the system reaches a stable equilibrium, the *harmonic* mean of the distance between any pair of nodes x, y belonging to an arbitrary bipartition U, V is inversely proportional to the density of the bipartition $d_{U,V}$:

$$\text{harmdist}(U,V) = |U| \cdot |V| / \left(\sum_{x \in U,\, y \in V} 1 / ||p_x - p_y||\right) = 1 / d_{U,V}$$

Because the harmonic mean distance weights small distances much higher than large distances, this energy models is particularly useful when drawing graphs. In fact, the well known graph visualization toolkit Gephi[6] uses LinLog to produces nice graph layouts (Figure 2).

QuadLin energy model. The QuadLin energy model is defined by the formula

6 Gephi, an open source graph visualization and manipulation software, https://gephi.org/.

$$E = \sum_{x,y \in E(U,V)} \tfrac{1}{2} \|p_x - p_y\|^2 - \sum_{x \in N(G),\, y \in N(G),\, x \neq y} \|p_x - p_y\|$$

where symbols have the usual meaning. QuadLin makes the *arithmetic* mean of the distance between any pair of nodes of an arbitrary bipartition inversely proportional to the density of the bipartition $d_{U,V}$:

$$\text{arithdist}(U,V) = \sum_{x \in U,\, y \in V} \|p_x - p_y\| / (|U| \cdot |V|) = 1 / d_{U,V}$$

At the equilibrium, nodes belonging to the same cluster tend to stay very close, much closer than in any other clustering energy model. This makes QuadLin particularly appealing for solving our original problem. In fact, the inverse of the distance between two nodes could be used a measure of the strength of their connection; which – in the case of an equivalence network – can be taken as an indication of the likelihood of their equivalence.

The previous equation could be written as

$$\sum_{x \in U,\, y \in V} \|p_x - p_y\| = (|U| \cdot |V|)^2 / |E(U,V)|$$

which should hold for any possible bipartition of the original graph, i.e. it is an over-determined system of linear equations which, in principle, could be solved using the linear least squares approach. Unfortunately, the total number of possible bipartitions of a graph G is the cardinality of the power set of N(G), $2^{|N(G)|}$, minus two (the empty set and N(G) itself). Thus, the most efficient existing way of solving such a system for non trivial graphs is by simulation.

4 Implementation

To illustrate the effectiveness of clustering energy model when applied to our original problem, we implemented a simple demonstrator system. The system is made up of a front-end and a back-end (Figure 1).

The front-end client runs in a Web browser as an AJAX application. After the user specifies a reference to a corresponding resource, the client connects to the back-end server, retrieves all known equivalences for that resources and displays them in the form of a 2D animation. Equivalent resources are presented as floating bubbles (Figure 2). The size of each bubble is proportional to its degree, i.e. the number of equivalences for the corresponding resource. The position of a bubble is finally determined by the composition of forces acting on it, forces derived from the QuadLin energy model. Their relative positions provide an indication of the strength of the equivalence. The bubbles also react to pointing device gestures, as if they altered their stable equilibrium. Clusters may be easily detected by looking at the different sets of nearby bubbles.

The back-end server performs two functions: retrieving equivalences from a RDF triplestore and providing a corresponding graph model, which, serialized, is sent to the client. The triplestore acts a cache server for RDF data extracted from Sindice.

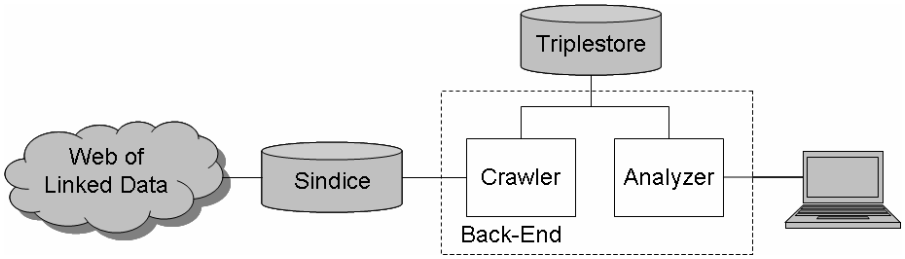

Figure 1. System architecture. The system is made up of a front-end, running in a Web browser, and a back-end, implemented as a Java 2 Enterprise application.

The back-end is implemented as a Java 2 Enterprise application running on Apache Tomcat 6.0 servlet container. The triplestore is Aduna Sesame server v2.60 powered by Gentoo Linux MySQL server v5.1. All the server-side software is hosted into a i686 Intel Xeon CPU 3060 machine, 2.40GHz processor, 1,048,772kB RAM, featured Gentoo Linux Base System 1.12 OS, kernel 2.6.18-xen.

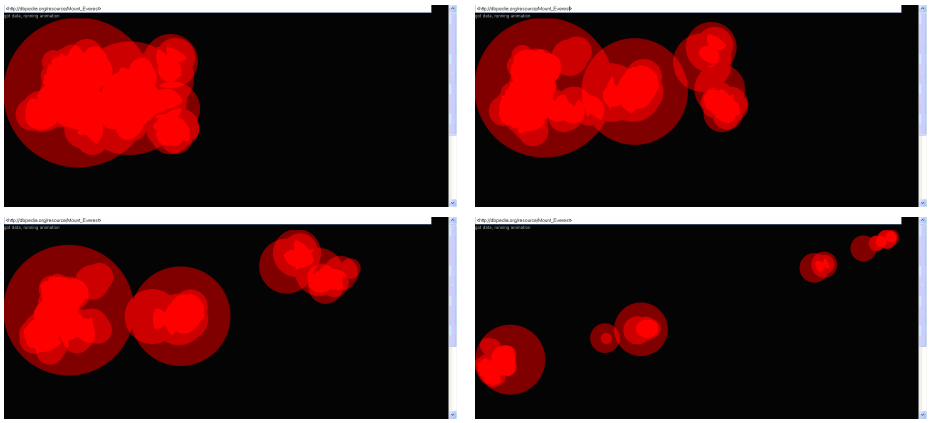

Figure 2. Front-end. Equivalent resources presented as floating bubbles. The size of each bubble is proportional to the number of its equivalence links. The position is determined by the composition of forces acting on it, according to the QuadLin energy model.

The client part is implemented using Google Web Toolkit, and is suitable to run on a variety of HTML 5 enabled browsers. The client runs a porting of the program originally developed by Noack[7]; but, differently from the original software, the porting uses the QuadLin energy model.

7 The original program by Noack is available at http://code.google.com/p/linloglayout/

5 Evaluation

Figure 3 shows the equivalence network obtained starting from the resource <http://rdf.freebase.com/ns/en.mount_everest>. We generated this picture using the Gephi framework which adopts the LinLog energy model. In this representation, the harmonic (not the arithmetic) mean of the length of the edges connecting two clusters is proportional to the inverse of the inter-cluster density; this feature provides a more readable layout, suitable, e.g., for this article's format, but reduces the inter-cluster nodes distance and increase the intra-cluster one. If QuadLin were used instead, nodes in the same cluster would be much closer and clusters more distant, as in Figure 2. Through the use of colours, Figure 3 also illustrates clusters as detected by Newman and Girvan's algorithm (section 2): nodes with different colours belong to different clusters.

Inspecting this figure, we see that LinLog (as well as QuadLin) recognizes the star-like topology around the resource <http://rdf.freebase.com/ns/en.mount_everest> and places all nodes belonging to this structure closer each others (bottom left of the picture). Newman and Girvan's algorithm fails in recognizing this topology, probably because of the small clustering coefficient[8] it presents: nodes and links in the picture are in fact marked with dozens of different colours, each meaning a potential different cluster. The star-like structure, also reported in [7], reflects the asymmetric use of owl:sameAs by the Freebase community: their nodes link many resources in DBpedia that however do not represent the very same concept (and in fact are not interlinked with each other, lowering the cluster coefficient).

A second structure is detected in the middle of the picture. This structure develops around the DBpedia resource <http://dbpedia.org/resource/Mount_Everest> and connects nodes belonging to a number of other domains. This structure has indeed a higher cluster coefficient and is correctly recognized by all the algorithms (LinLog, QuadLin and Newman and Girvan's algorithm).

The top right corner of the picture clearly shows a third cluster, connected to the previously described one by one single arc. The cluster is well separated from the other two, although not reflected by LinLog – the same distance measured out after applying the QuadLin algorithm is about 130% of the distance between the two previous structures. This is not surprising: because of one single arc connecting the two structures, the density of the corresponding bipartition is small and the nodes on the two sides are far. In our method, a great spatial distance implies a marked semantic difference. Looking at this third cluster, we realize that all its nodes refer to a different entity: Sun Valley in Blaine County, Idaho, US.

8 The clustering coefficient is a local property of a node and is defined as the fraction between the average number of edges between the neighbors of a node and the average number of possible edges between these neighbors.

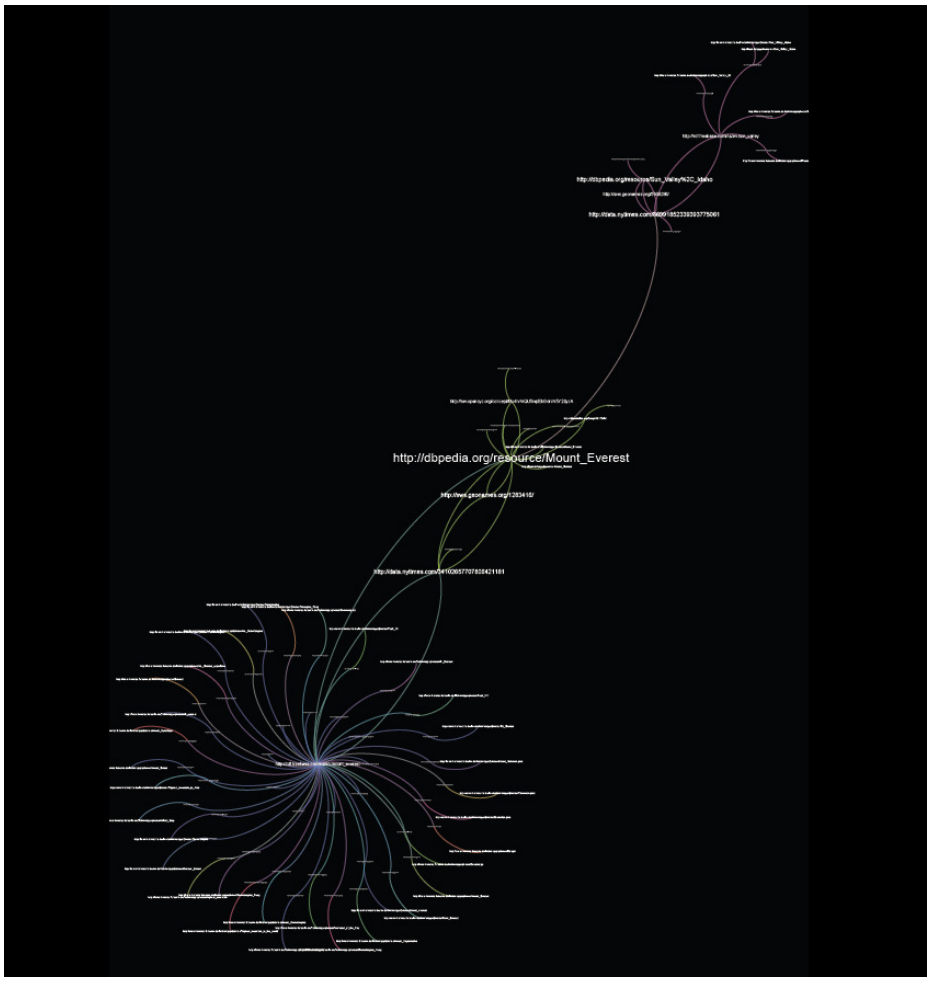

Figure 3. The equivalence network for the entity “Mount Everest” (picture generated using the Gephi framework which adopts the LinLog energy model).

6 Discussion (and Answers to Questions)

Question #1 – (from section 1) Where could a possible weighted uncertainty measure of identity come from? Experimental evidences show that co-references coming from different datasets show a tendency to aggregate into star-like structures, and more in general into clusters. Missing intra-cluster arcs can be probably interpreted as omitted equivalences and isolated inter-cluster arcs are likely to represent wrongly stated equivalences. To distinguish the two case, we assume two

thresholds, m and M, run a simulation, and calculate the edge length $||p_x - p_y||$ between each pair of connected nodes (x,y). If $||p_x - p_y|| < m$, then nodes are in the same cluster and likely to be equivalent; if $||p_x - p_y|| > M$, probably the two nodes are different from each other and would be better characterized by the owl:differentFrom relationship.

The difference between clusters and non-clusters lies purely in the density of the links connecting a given group of nodes, and in the density of the links connecting this group with the rest of the graph. This "vague" definition does not provide any canonical value to distinguish a cluster from a weaker group of nodes. Therefore the aforementioned thresholds have to be evaluated for each equivalence network. Are there advantages in shifting the original dilemma of evaluating the strength of a single equivalence link to the (maybe as difficult) problem of finding global thresholds for the equivalence network the link belongs to? We think so. First, a metric has been defined, enabling ranking of the strength of different equivalence links. Second, statistics could help in defining canonical or typical values like thresholds. We expect to mine these values from massive batch processing we are currently carrying on.

Question #2 – (from section 2) Why do nodes aggregate into clusters? In our interpretation, this phenomenon typically reflects the partition of linked data publishers into communities, where nodes representing similar concepts inside the two different communities of publishers are loosely coupled. The communities may have different purposes, degree of specialization and ways of defining equivalences. For instance, DBpedia provides lots of hyponyms (i.e. more specialized terms, finer granule definitions) while other data providers – including Freebase – publish more generic definitions. Nodes from Freebase usually link hyponyms in DBPedia, semantic nuances are not captured during the linkage and are flattened as "equivalences" with the corresponding hyperonym (i.e. the more generic tem). Though a mistake when considering the formal semantics, this way of establishing equivalences is not always annoying; sometimes it is even helpful end users who might want to follow links to discover slightly different meanings and thus increase their knowledge about a subject.

On the other hand, the raising of new communities and addition of new nodes and links in the future might create new specialized clusters "detached" from the parent ones and reflecting hyponyms more closely. This phenomenon probably mirrors the way knowledge specializes, beginning from a rough level of definition and getting structured when interests from members of domain-specific communities start developing. Cluster detection techniques may provide effective insights for this investigation.

7 Conclusions

Equivalence links connecting co-references may be seen as a graph known as "equivalence network". In this paper we introduced a method to rank equivalence links based on the contextual knowledge of the topology of this graph. The rank provides an estimation of the strength of each equivalence link. The method can be used either in batch processing mode (suitable to provide analysis of massive datasets and to extract canonical values) or interactively; here we presented a simple prototype

which exploits the latter option. As a tool, it may help Linked Data engineers to better understand and "debug" equivalence networks and, indirectly, to unveil different emerging communities of publishers, their different goals and linking strategies.

References

1. Hu, W., Qu, Y. and Sun, X.: Bootstrapping object coreferencing on the semantic web. Journal of Computer Science Technology, 26(4), 663–675, 2011.
2. Bizer, C., Health, T.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory & Technology, 1:1,1-136. Morgan Claypool, 2011.
3. Halpin, H., Hayes, P.P., McCusker, J., Mcguinness, D., Thompson, H.: When owl:sameas isn't the same: An analysis of identity in linked data. In Proceedings of the 9th International SemanticWeb Conference, 2010. DOI: 10.1007/978-3-642-17746-0_20 23
4. McCusker, J., McGuinness, D., owl:sameas Considered Harmful to Provenance. In Proceedings of the ISCB Conference on Semantics in Healthcare and Life Sciences, 2010
5. Hayes, P., Halpin, H., In defense of ambiguity. International Journal of Semantic Web and Information Systems, 4(3), 2008.
6. Ding, L., Shinavier, J., Finin T., McGuinness, D.: owl:sameAs and Linked Data: An Empirical Study. Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, 2010.
7. Ding, L., Shinavier, J., Shangguan Z., McGuinness, D.: 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, 2010.
8. Bartolomeo, G., Salsano, S.: A Spectrometry of Linked Data. In Proceedings of the Linked Data on the Web Workshop 2012.
9. Strogatz, S. H.: Exploring complex networks. Nature, 410, 268–276, 2001.
10. Newman, M. E., Girvan, M.: Finding and evaluating community structure in networks. In Physical Review E, volume 69, issue 2, 2004.
11. Noack, A.: Energy Models for Graph Clustering, Journal of Graph Algorithms and Applications, Vol. 11, no. 2, pp. 453-480, 2007.
12. Jaffri, A., Glaser, H., Millard, I.: URI disambiguation in the context of linked data. In Proceedings of the 1st International Workshop on Linked Data on the Web, 2008.
13. Halpin, H., Presutti, V. 2009. An ontology of resources: Solving the identity crisis. In: ESWC 2009, Research Studies Press/Wiley.
14. 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, ICSC 2008 554-561 IEEE Computer Society.
15. 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.