Analyzing Linked Data Quality with LiQuate

Edna Ruckhaus, Oriana Baldizán, and María-Esther Vidal

Universidad Simón Bolívar, Venezuela
{eruckhaus,obaldizan,mvidal}@ldc.usb.ve

Abstract. In the last years, the number of datasets in the Linking Open Data (LOD) cloud and the applications that rely on links between these datasets to discover patterns or potential new associations, have exploded. However, because of data source heterogeneity, published data may suffer of redundancy, inconsistencies or may be incomplete; thus, results generated by linked data based applications may be imprecise or unreliable. We illustrate LiQuate (Linked Data Quality Assessment), a tool that combines Bayesian Networks and rule-based systems to analyze the quality of data and links in the LOD cloud.

1 Introduction

Linking Open Data initiatives have made a diversity of collections available, and facilitate scientists the mining of linked datasets to discover patterns or suggest potential new associations. To ensure trustworthy results, linked data must meet high quality standards; however, data in the LOD cloud has not been necessarily curated, and tools are required to detect possible ambiguities and quality problems. To achieve this goal, we developed LiQuate, a semi-automatic tool able to identify ambiguities among the linked data, and suggest possible inconsistencies and incompleteness. LiQuate implements a two-fold approach that combines Bayesian Networks and rule-based systems to analyze the quality of data and propose new links to resolve the identified ambiguities. First, a Bayesian Network models dependencies among resources in a set of linked datasets [4]; conditional probability tables annotate the nodes of the network and represent joint probability distributions of relationships among resources. Queries against the Bayesian Network represent the probability that different resources have redundant labels or that a link between two resources is missing; thus, the returned probabilities can suggest ambiguities or possible incompleteness in the data or links. Second, a probabilistic rule-based system is used to infer new links that associate equivalent resources, and allows to resolve the ambiguities and incompleteness identified during the exploration of the Bayesian Network.

We demonstrate the data quality validation capabilities of LiQuate and the benefits of the approach on the Life Science datasets: *LinkedCT*[1], *Diseasome*[2], *Drugbank*[3], and *DBPedia*[4] datasets. We show the following key issues:

[1] `http://www.cs.toronto.edu/~oktie/linkedct/` downloaded Sept 2011.
[2] `http://datahub.io/dataset/fu-berlin-diseasome` downloaded Sept 2011.
[3] `http://datahub.io/dataset/fu-berlin-drugbank` downloaded Sept 2011.
[4] `http://wiki.dbpedia.org/Downloads32` downloaded Sept 2012.

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 629–638, 2013.

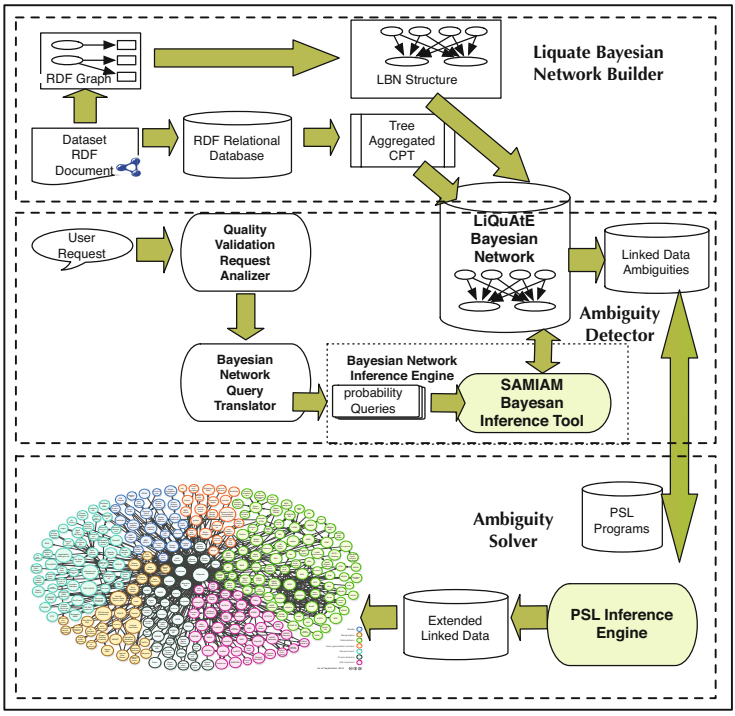

Fig. 1. The LiQuiate System Architecture

redundancy of labels that correspond to drugs in *LinkedCT* and *Drugbank*, and to diseases in *LinkedCT* and *Diseasome*, and incompleteness and inconsistencies of links between these Life Science datasets. The demo is published at `http://liquate.ldc.usb.ve`.

The structure of this paper is as follows: In Section 2 we present the Liquate System Architecture and components. Following this section we present the formalization of our approach, the Linked Bayesian Network (LBN). Next we present the related work and our experimental study, and finally, section 6 concludes and outlines interesting future directions.

2 The LiQuate System

LiQuate implements a two-fold approach where first, ambiguities are detected and then, links to solve these ambiguities are inferred and suggested to the user for resolving the identified quality problems; Figure 1 illustrates the LiQuate architecture. Currently, three types of quality validation requests can be expressed: *i*) probability that labels or names of a given (type of) resource are redundant, *ii*) probability of incomplete links among a given set of resources, and *iii*) probability of inconsistent links. LiQuate is comprised of three components: the `LiQuate Bayesian Network Builder`, the `Ambiguity Detector` and the `Ambiguity Solver`. The `LiQuate Bayesian Network Builder` is a semi-automatic off-line process; it relies on an expert's knowledge about the

properties in the RDF linked datasets that are going to be represented in the Bayesian Network. Relevant data is retrieved from SPARQL endpoints, and stored in a relational database to compute the histograms that implement the conditional probability tables (CPTs) associated with the nodes of the network. The `Ambiguity Detector` is a probabilistic model that supports the analysis of the above mentioned linked data quality problems. The `Ambiguity Detector` is in turn comprised of three components: *1*) the `Quality Validation Request Analyzer`, *2*) the `Bayesian Network`, `Query Translator`, and *3*) the `Bayesian Network Inference Engine`. The `Quality Validation Request Analyzer` receives a user request and decides if it can be satisfied with the existing Bayesian Network. The `Bayesian Network Query Translator` considers the user request and generates the set of queries that must be posted against the Bayesian Network. It also gathers the answers of these queries and answers the user request. Finally, the `Bayesian Network Inference Engine` is responsible of performing the inference process required to answer each of the queries posted against the Bayesian Network; this engine is implemented by the *SamIam* Bayesian Inference Tool[5].

The `Ambiguity Solver` is a rule-based system that infers new links that solve the ambiguity and incompleteness problems identified by the `Ambiguity Detector`. The rule-based system is implemented as a set of probabilistic rules [9] that reason about similarity to generate additional links with a certain degree of uncertainty. These new links may associate resources to control resource redundancy, or to control link inconsistency. The Ambiguity Solver has been implemented on top of the Probabilistic Soft Logic tool (PSL)[6]. The PSL Inference Engine receives a set of weighted rules, and infers with a certain degree of uncertainty when two terms are duplicates, incomplete or incorrect. PSL similarity metrics are implemented to decide when two labels are similar. Inferred data quality problems between two terms are used to suggest RDF intra- and inter-links. The output of this component is an RDF document comprised of `owl:sameAs` links that relate redundant resources; the quality of the new generated links can be determined by experts or against a given ground truth.

To conclude, the conditional probabilistic approach is used to suggest possible ambiguities in linked datasets, while the PSL approach is just used to suggest new links for ambiguity resolution. PSL allows to infer with a certain degree of uncertainty if a given new link can be included to resolve an ambiguity identified by the `Ambiguity Detector`, and this degree can be configured. Thus users can decide the appropriate level of uncertainty for a given domain and assign different ground truths to analyze the quality of the new links. Additionally, in the current version of LiQuate, the `Ambiguity Solver` does not perform any prediction task; it just uses information inferred from the Bayesian network to suggest with a certain degree of uncertainty, a link between two possible redundant concepts. However, it is important to highlight that PSL features can be also exploited to implement prediction techniques that rely on graph analysis algorithms, to

[5] `http://reasoning.cs.ucla.edu/samiam/help/recursiveconditioning.html`
[6] `http://psl.umiacs.umd.edu/`

determine the density of the linked datasets and based on this, suggest potential missing links. This problem is out of the scope of this paper, and we mainly focus on the `Ambiguity Detector` component.

Finally, Figure 2 presents the workflow that is followed for the construction of the LBN CPTs. The steps are the following:

1. RDF datasets are loaded as vertically partitioned relational tables [1] (one table per property with subject and object columns).
2. A CPT for root nodes `s-<property>` and `o-<property>` is created in O_B through queries to the property tables which count and group the different object or subject values.
3. A CPT for link nodes, `b-<linkprop>-<typeres1>-<typeres2>` is created through queries to the link property tables which count and group the different linked resources.
4. A CPT for join nodes, `s-s-<property1>-<property2>` and other join nodes is created through join queries to the property tables and their parent property tables which count and group the parent object/subject values.
5. All CPT tables are ordered by their probability value, and the frequency histogram is generated. Some auxiliary data structures have been created in order to speed-up the lookup of the CPT histogram values: (1) the *Corr* structure that registers the correspondence of each CPT value with a sequential number, and (2) the *CptIndex* structure for non-root nodes, where for each sequential number representing a parent CPT value, there is a reference to the corresponding entry in the parent's aggregated histogram.

3 The Liquate Bayesian Network

A Liquate Bayesian Network (LBN) is a probabilistic model of a network of linked RDF datasets. It represents all the conditional dependencies among property subjects and objects in single RDF datasets and in linked datasets. The analysis of these dependencies is used to detect linked data quality problems.

The LBN model is based on the Bayesian network model developed for relational domains in [4], the Probabilistic Relational Model (PRM). Although an LBN resembles a PRM, its nodes and arcs have a particular semantics based on the linked RDF graph semantics. Nodes represent property objects and subjects, and intra- and inter-links. Figure 3 presents the LBN for RDF datasets **LinkedCT**, **Diseasome**, **Drugbank** *circa* September 2010.

Definition 1 (LiQuate Bayesian network). *Given an RDF directed graph $O_R = (V_R, E_R)$ where V_R, and E_R are the nodes and arcs in the RDF graph. A* **LiQuate Bayesian network** *R_B for O_R, is a pair $R_B = \langle O_B, CPT_B \rangle$, where $O_B = (V_B, E_B)$ is a DAG. V_B are the nodes in O_B, and E_B are the arcs in O_B, and an homomorphism $f : \mathbb{P}(E_R) \Rightarrow \mathbb{P}(V_B)$ establishes a mapping between the power set of sets of graph edges (sets of triples) in O_R and sets of nodes in O_B. CPT_B are the Conditional Probability Tables for each node.*

There are three types of nodes:

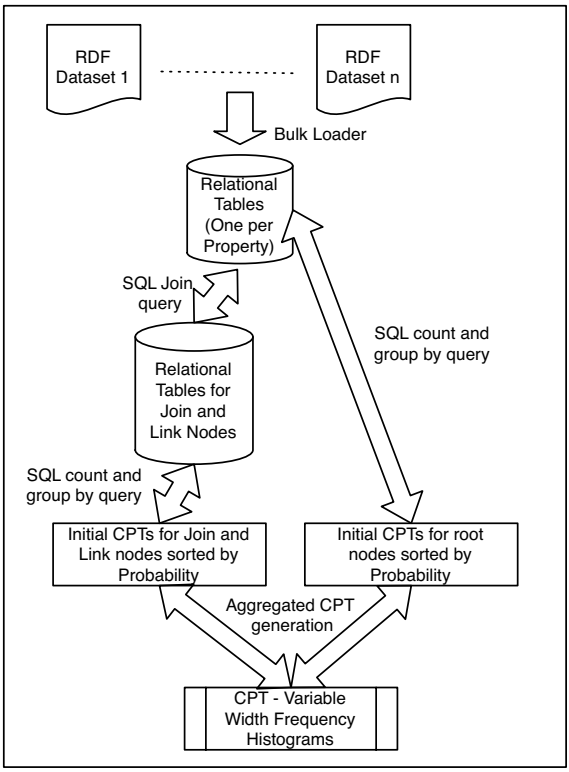

Fig. 2. A Workflow to Build an LBN Given Several RDF Datasets

1. **Value** nodes: `s-<property>` and `o-<property>` represent property subjects or objects in a single dataset. For example, node `o-hasintervention` represents the object values of interventions in clinical trials.
2. **Join** nodes: `s-s-<pro1>-<pro2>`, `o-s-<pro1>-<pro2>` and `o-o-<pro1>-<pro2>` correspond to boolean variables, and represent the matching of subjects or objects in related properties in a single dataset.
   For example, node `s-s-hascondition-hasintervention` represents the "join" over a trial, that is, a condition and an intervention are part of a trial.
3. **Link** nodes: `b-<linkprop>-<typeres1>-<typeres2>` corresponds to a boolean variable, and represents the existence of links among related resources. For example, node `b-sameas-condition-disease` represents the existence of *owl:sameAs* links among conditions in Clinical Trials (LinkedCT) and diseases in Diseasome.

The first two types of nodes represent data items and intra-dataset links [12], respectively, while the third type of nodes corresponds to inter-dataset links. Arcs represent dependencies between nodes. The event represented by node `s-s-hascondition-hasintervention` is conditioned by the values of condition, intervention (nodes `o-hascondition` and `o-hasintervention`), by the existence of

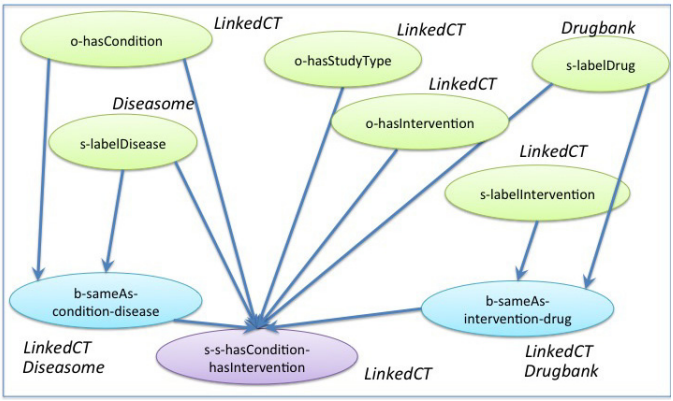

Fig. 3. LiQuate Bayesian network for the Life Sciences Domain

an *owl:sameAs* link among the condition and a disease, and among the intervention and a drug (nodes `b-sameAs-condition-disease` and `b-sameAs-intervention-drug`). For each modeled dataset, there is a set of nodes annotated with the URI of the dataset. The basic LBN inference task is a marginal posterior probability query *prob(X|e)*, where the marginal variable X is represented by a **Join** or **Link** node, and where the evidence E is set up according to the particular query. LiQuate uses an exact inference algorithm: *Shenoy-Shafer*. For example, a query will check if given - the evidence - that the drug *Paclitaxel* has as possible disease target *Leukemia*, there is a clinical trial that backs this relationship with an equivalent - *owl:sameAs* - condition and intervention:

*prob(*s-s-hascondition-hasIntervention|
b-possibleDisease-drug-disease ∧ s-labelDrug='Paclitaxel' ∧
s-labelDisease='Leukemia' ∧ b-sameAs-condition-disease ∧
b-sameAs-intervention-drug)

The answer to this probability query is 1.0; this result indicates that this relationship is backed by a clinical trial.

The homomorphism $f : \mathbb{P}(E_R) \rightarrow \mathbb{P}(V_B)$ establishes a mapping between O_R and O_B. f defines the set of nodes V_B. The CPT_B are multidimensional histograms ordered by value. If a node v is a source node, the histogram will be one-dimensional, because in this case the CPT_B only represents the distribution of values taken up by the variable represented by the node. The size of a CPT for one node depends on the number of predecessors, and on the number of possible values for the node and its predecessors. In our case the structure of the Bayesian network is related to the number of properties in the RDF datasets; in general, the RDF datasets are not complex in terms of their classes and properties. As to the set of possible values, these were aggregated in an indexed histogram that represents the CPT values.

4 Related Work

The publication of clinical trials as linked RDF data is described in [6]. In this work, the authors emphasize the challenges of linking resources in the trials data, and linking different datasets. The authors demonstrated how state-of-the-art approximate string matching and ontology-based semantic matching can be used for discovery of such semantic links between several data sources. Differently from our work, the emphasis is on link discovery while LiQuate's focus is on the detection of linked data quality problems and the enrichment of data links.

The framework xCurator proposed by Hassas et al [7] aims to produce high quality linked RDF data from semi-structured sources where unique URIs are generated, duplicates are merged, resources are linked to vocabularies using entity type extraction techniques, and links are established to external RDF sources. The framework was applied to clinical trial and to bibliographic data, and the quality of the data was improved with respect to previous transformations that were done manually. This work is focused on the entity extraction component as the means to improve linked data quality; Contrary, LiQuate exploits semantics encoded in the Bayesian network during a statistical inference process, and is able to suggest possible ambiguities and inconsistencies not only by considering the names of the entities, but also by looking at the different entities that are related to the studied entities.

In [8], the applicability and benefits of using linked data in the Life Sciences domain is studied, specifically for clinical trials, drugs, and related sources. The authors present several challenges and among them, the need of progress in finding links between data items where no commonly used identifiers exist, and the need to develop techniques for record linkage and duplicate detection with methods from the database and knowledge representation communities. The challenges presented summarize some of the data quality problems that we detected in our experimental study.

Demartini et. al. [2] develop a probabilistic framework in combination with crowdsourcing techniques, in order to improve the quality of links in the LOD cloud. The system, *ZenCrowd* combines algorithmic and manual matching techniques to link entities. It exploits probabilistic models using factor-graphs to represent probabilistic variables. This approach is based on evaluating several alternative links, whereas our system proposes evaluating the quality of the current links in some specific domain.

Network approximate measures are used to analyze the quality of linked data in [5] using the LINK-QA framework. An original local network and extended networks are constructed around resources to be evaluated by querying the Web of Data. Five metrics are used. degree, clustering coefficient, number of *owl:sameAs* chains, centrality and richness of description. Fürber et. al. [3] propose a conceptual model for data quality management that allows to formulate a set of data quality and data cleaning rules, classification of data quality problems, and the computation of quality scores for Semantic Web data sources. The authors present several use cases and competency questions, but these are all related to the quality of the classes, properties and instances on the datasets,

but not in the consistency or quality of the links. A set of quality and cleaning rules is established, but none of these refer to links among datasets.

Memory et. al. [11] present a work on summarization of annotation graphs where PSL is the framework used. The work integrates the multiple types of evidence from the annotation links, the various similarity metrics, and two graph summarization heuristics: a similarity heuristic and a summarization heuristic within a probabilistic model, using PSL. This approach uses PSL to model the summarization graph whereas in our work, a PSL system is used to propose additional links with a certain degree of uncertainty.

5 Experimental Study

As of September 2011, LinkedCT contains 106,308 trials, 2.7 million entities and over 25 million RDF triples. Additionally, we consider the following datasets that are linked to LinkedCT: *i*) Drugbank (over 765,936 triples), *ii*) Diseasome (around 91,182 triples), and *iii*) DBPedia (links from LinkedCT 25,476). We built local RDF storage with LinkedCT triples and the triples from these three datasets that are related to LinkedCT. The Bayesian network and its corresponding CPT's were computed and stored in the *SamIam* Bayesian Inference Tool. The generated network is comprised of 17 nodes and the aggregated CPTs are of up to $167,616$ entries; for the cases to be shown, the average response time of LiQuate is $4,715$ ms.

For each dataset, we partition the entities according to their label, e.g., diseases are partitioned according to their name. The metrics that are considered in the experimental study are % of partitions with size > 1, % partitions with at least one *owl:sameAs* (resp. *rdfs:seeAlso*) link, and % partitions with all labels with *owl:sameAs*(resp. *rdfs:seeAlso*) links. We conducted the following studies:

Ambiguities between labels of Interventions or Drugs: starting with Alemtuzumab as an exemplar, we retrieve the intersection of Monoclonal antibodies and Antineoplastic agents. This creates a dataset of 12 drugs: Alemtuzumab, Bevacizumab, Brentuximab vedotin, Cetuximab, Catumaxomab, Edrecolomab, Gemtuzumab, Ipilimumab, Ofatumumab, Panitumumab, Rituximab, and Trastuzumab. These drugs are frequently tested in clinical trials, and there are up to 723 clinical trials for a given drug or intervention, e.g., the intervention that corresponds to the drug Alemtuzumab is associated with 112 different clinical trials. We can observe that 15.74% of partitions have a size > 1, i.e., are redundant.

Incompleteness of links between LinkedCT, Drugbank, Diseasome and DBPedia: We will consider the family of the 12 drugs and for each of the partitions induced by duplicated labels, we consider the `owl:sameAs` and `rdfs:seeAlso` links. Some cases are: a percentage of redundant labels is not linked through `owl:sameAs` to neither Drugbank or DBPedia, but 100% of the labels are linked through `rdfs:seeAlso`, e.g., Bevacizumab; none of the redundant labels is linked to Drugbank or DBPedia, e.g., Brentuximab vedotin; in this case, the drug is not present in Drugbank; all of the redundant labels are linked to DBPedia and none to Drugbank, e.g., Catumaxomab; a percentage of redundant labels

is linked to DBPedia through `owl:sameAs`, all of them are linked to DBPedia through `rdfs:seeAlso` and none to Drugbank, e.g., Ipilimumab.

Inconsistencies of links between LinkedCT, Drugbank, Diseasome and DBPedia: We will analyze if the relationships that represent that a disease is a possible target of a drug and if a drug can target a disease, are supported by a clinical trial, i.e., diseases targeted by a drug with a link *possibleDiseaseTarget*, are supported by at least one clinical trial with a condition and drug intervention, and vice versa. Approximately 10,000 probability queries were generated for each drug and disease and all the combinations of linked (through `owl:sameAs`) conditions and interventions. The marginal node is `s-s-hascondition-hasintervention` (violet node in Figure 3), and the evidence is a disease, drug, condition, intervention and the existence of `owl:sameAs` links among them. The result is that only 13,5% of the drugs and targeted diseases are supported by clinical trials that can be found through `owl:sameAs` links. Similarly, another hypothesis is that drugs that can possibly treat diseases (*possibleDrug* links) are supported by the same number of clinical trials. The result is 13, 5% and this number suggests that both links *possibleDiseaseTarget* and *possibleDrug* are the inverse of each other. Particularly, for the dataset of 12 drugs we can observe the following: the drugs Brentuximab vedotin, Ipilimumab and Ofatumumab do not appear in Drugbank while these drugs have been studied in a large number of clinical trials. The rest of these 12 drugs do appear in Drugbank, but are associated with much less diseases through the property *possibleDiseaseTarget* in Drugbank, than to conditions through a clinical trial in LinkedCT; e.g., the drug Cetuximab can possibly target eighteen diseases while this drug has been tested in completed clinical trials for 82 conditions; only four of the eighteen diseases in the property *possibleDiseaseTarget* in Drugbank, are included in the list of 82 conditions in LinkedCT. This ambiguity can be also observed in the rest of the drugs.

6 Conclusions and Future Work

We present LiQuate, a data and link validation tool that relies on a Bayesian Network to identify redundancies, incompleteness and inconsistencies, and makes use of a probabilistic rule-based system to infer the links that solve the identified quality problems with a certain degree of uncertainty. We demonstrate the main quality validation capabilities of LiQuate, and illustrate different quality problems that may currently occur in the LOD cloud. Particularly, we can observe some ambiguities that suggest the experts to check for uncontrolled redundancy, incompleteness or inconsistency: *i*) the same label or name of intervention is assigned to different resources, *ii*) incomplete `owl:sameAs` and `rdfs:seeAlso` links between datasets, and *iii*) associations between drugs and diseases in Drugbank may be not supported by trials in LinkedCT. Provenance information of the datasets will be taken into account in order to not only enrich the datasets with the solution to the quality problems, but to be able to modify them.

Furthermore, the generation of the CPT tables for the LBN can be developed using the RDFStats[10] statistics generator, or use their API for accessing

statistics including several estimation functions that also support SPARQL filter-like expressions. Finally, experimental studies will be developed for other domains in order to do a thorough evaluation of the system and validate the proposed links. Particularly, we plan to validate links to terms of Geonames[7].

References

1. Abadi, D.J., Marcus, A., Madden, S.R., Hollenbach, K.: Scalable semantic web data management using vertical partitioning. In: Proceedings of VLDB 2007 (2007)
2. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: WWW (2012)
3. Fürber, C., Hepp, M.: Towards a vocabulary for data quality management in semantic web architectures. In: EDBT/ICDT Workshop on Linked Web Data Management (2011)
4. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. SIGMOD Record 30(2), 461–472 (2001)
5. Guret, C., Groth, P., Stadler, C., Lehmann, J.: Linked data quality assessment through network analysis. In: ISWC 2011 Posters and Demos (2011)
6. Hassanzadeh, O., Kementsietsidis, A., Lim, L., Miller, R.J., Wang, M.: Linkedct: A linked data space for clinical trials. CoRR, abs/0908.0567 (2009)
7. Hassanzadeh, O., Yeganeh, S.H., Miller, R.J.: Linking semistructured data on the web. In: WebDB (2011)
8. Jentzsch, A., Andersson, B., Hassanzadeh, O., Stephens, S., Bizer, C.: Enabling Tailored Therapeutics with Linked Data. In: Proceedings of the WWW 2009 Workshop on Linked Data on the Web (LDOW 2009) (2009)
9. Kimmig, A., Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: A short introduction to probabilistic soft logic. In: NIPS Workshop on Probabilistic Programming: Foundations and Applications (2012)
10. Langegger, A., Wöß, W.: Rdfstats - an extensible rdf statistics generator and library. In: DEXA Workshops (2009)
11. Memory, A., Kimmig, A., Bach, S.H., Raschid, L., Getoor, L.: Graph summarization in annotated data using probabilistic soft logic. In: URSW (2012)
12. Ruckhaus, E., Vidal, M.-E.: The BAY-HIST Prediction Model for RDF Documents. In: Proceedings of the 2nd ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web-CEUR, vol. 611, pp. 30–41 (2010)

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