Demo paper at the 11th International Semantic Web Conference (ISWC'12), Boston USA, Nov 2012.

Demonstrating Blank Node Matching and RDF/S Comparison Functions

Christina Lantzaki, Yannis Tzitzikas, and Dimitris Zeginis*

Institute of Computer Science, FORTH-ICS, GREECE, and
Computer Science Department, University of Crete
{kristi,tzitzik,zeginis}@ics.forth.gr

This demo paper accompanies the ISWC'2012 paper: [7]

Motivation

The ability to compute the differences that exist between two RDF/S Knowledge Bases (for short KBs) is important for aiding humans to understand the evolution of knowledge, and for reducing the amount of data that need to be exchanged and managed over the network in order to build SW synchronization, versioning and replication services [2, 3, 1, 8, 6].

A rather peculiar but quite flexible feature of RDF is that it allows the representation of *blank nodes*: a *blank node* (or anonymous resource or bnode) is a node in an RDF graph which is not identified by a URI and is not a literal. Several KBs rely heavily on blank nodes as they are convenient for representing complex attributes (e.g. an attribute `address`) without having to name explicitly the auxiliary node that connects together the values that constitute the complex value (e.g. the particular `street, number` and `postal code` values). Bnodes are also convenient for resources whose identity is unknown but their attributes (either literals or associations with other resources) are known. According to [4], blank nodes is an inevitable reality, e.g. the data fetched from the "hi5.com" domain consist of 87.5% of blank nodes.

The inability to match bnodes increases the delta size and does not assist in detecting the changes between subsequent versions of a KB [4].

Approach

Although there are several works on blank node and comparison functions (for details see [7]), the problem has not been thoroughly studied. To the best of our knowledge, the only work that attempts to establish a bnode mapping for reducing the size of deltas also for the case of non equivalent KBs is [7] (ISWC'12). Finding such a mapping can be considered as a preprocessing step, a task that is carried out before a differential function is applied.

In [7] we prove that finding the optimal mapping is NP-Hard in the general case, and polynomial if there are no directly connected bnodes. Subsequently,

* Current affiliation: Information Systems Lab, University of Macedonia, Thessaloniki, Greece, zeginis@uom.gr

we present two main algorithms: (a) the Alg_{Hung} algorithm, and (b) the Alg_{Sign} algorithm. Alg_{Hung} solves the optimization problem using the *Hungarian algorithm* [5], an algorithm for solving the *assignment problem*. For the cases where there are directly connected bnodes, a variation of Alg_{Hung} is used for producing an *approximate* solution. The time complexity of the Alg_{Hung} in any case is in $O(n^3)$, where n is the number of bnodes.

For making the application of this method feasible also to very large KBs, at the cost of probably bigger deltas, [7] also proposed a *signature*-based method, Alg_{Sign}, whose complexity is in $O(n \log n)$. For these algorithms, the reported experimental results over real and synthetic datasets showed significant reductions of the sizes of the computed deltas.

What will be Demonstrated

We will demonstrate a tool called `BNodeDelta` which supports all algorithms presented at [7]. With this tool, the user (human or other program), specifies the two KBs to be compared (which can be stored in local files or fetched from the network using HTTP), then specifies the bnode mapping algorithm to be used, and then gets back statistics (about the KBs and their delta) and the delta itself (sets of triples to be added and deleted). Furthermore the tool can take as input a *namespace mapping table* (if a namespace $nm1$ is mapped to a $nm2$ then they are considered equal at the comparison phase).

We will demonstrate the system using two real datasets available in the LOD cloud: the *Swedish open cultural heritage* dataset[1], and the *Italian Museums* dataset[2], published from LKDI[3]. We shall also use synthetically generated data.

Figure 1 (left) shows the command line interface which shows the basic statistics for the Italian dataset (more statistics can be placed on demand in a file called "statistics"). Figure 1 (right) shows an excerpt of the file that contains the added triples (assuming the user requested the output *delta* in RDF/XML format).

We will give emphasis on the bnode mapping algorithms, specifically we will show the size of the outcome of the differential function Δ_e (where $\Delta_e(K \rightarrow K') = \{Add(t) \mid t \in K' - K\} \cup \{Del(t) \mid t \in K - K'\}$) for the cases: Alg_{Hung}, Alg_{Sign}, a random bnode mapping algorithm, and no bnode mapping at all.

Time Efficiency (comparative results). In both algorithms (Alg_{Hung} and Alg_{Sign}) the required time depends on the number of bnodes of the two KBs and the average number of triples to which a bnode participates. Alg_{Hung} needs 5.4 seconds over datasets of average $3,650$ triples and 525 bnodes, and 9.6 minutes for datasets of average $49,900$ triples and $6,390$ bnodes, whereas Alg_{Sign} needs only 0.34 seconds and 0.92 seconds respectively. These results show that Alg_{Sign}

[1] http://thedatahub.org/dataset/swedish-open-cultural-heritage used from http://kringla.nu/kringla/ for providing information on cultural data of Sweden
[2] http://thedatahub.org/dataset/museums-in-italy
[3] http://www.linkedopendata.it/

can be efficient also in bigger datasets (we will also show that two KBs with 153,600 bnodes can be compared at less than 11 seconds).

Fig. 1. Basic statistics and added triples of delta over Italian datasets

Delta Sizes (comparative results). As regards delta size, in the first dataset without bnode mapping the delta contains 5,771 triples, whereas with Alg_{Hung} it contains 311 triples, and with Alg_{Sign} 419 triples.

In the second dataset without bnode mapping the delta contains 43,770 triples, whereas with Alg_{Hung} it contains 6 triples, and same for Alg_{Sign}.

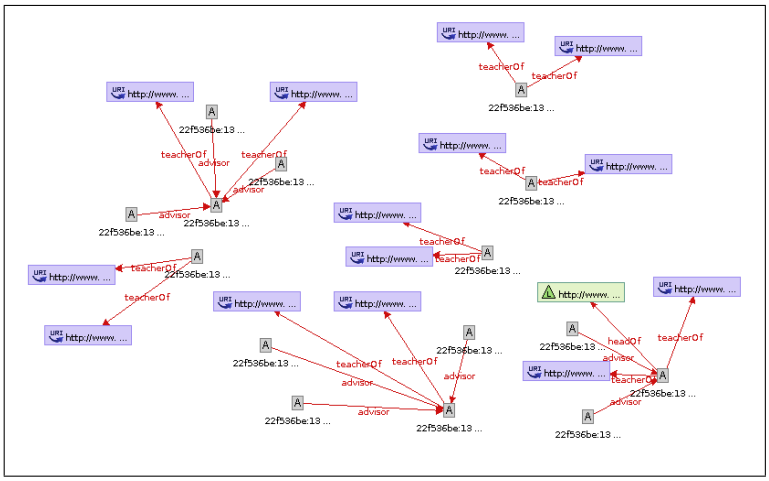

Fig. 2. Visualization of the added triples over synthetic datasets with RDF-Gravity

Delta Visualization. Apart from the benefits in a versioning/synchronization scenario, the achieved delta size reduction makes the visualization and exploration of the delta much easier. For this reason, `BNodeDelta` offers several choices for formatting the output delta in order to aid further processing or visualization. One option returns the delta in two separate files, one containing the *deleted* triples, the other the *added* triples, both in RDF/XML format. Each of these files can be explored and visualized with various RDF/S visualization tools.

For instance, we loaded to RDF-Gravity[4] the RDF/XML file that contains the *added* triples of the delta over the synthetic dataset. Figure 2 shows the derived visualization
Note that if the delta size is small, *both* added and deleted triples can be visualized as a single graph. In such cases, `BNodeDelta` also returns a graph visualization. For example, Figure 3 shows the graph of delta over the Italian datasets where the added elements are in *green* while the deleted are in *red*.

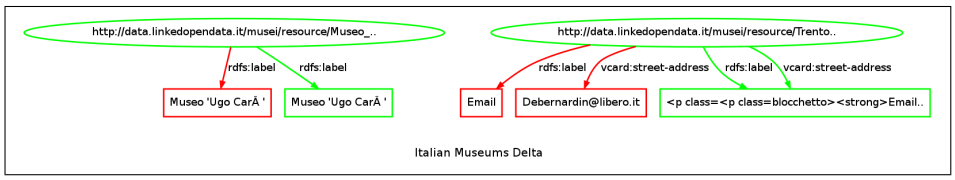

Fig. 3. A graph-based visualization of the delta over Italian datasets (in red the deleted triples, in green the added ones)

The above examples, just show that bnode mapping can reduce the delta to sizes appropriate for graph-based visualization (something not possible without bnode mapping).

Software and datasets are available to download and use from `http://www.ics.forth.gr/isl/BNodeDelta`.

References

1. T. Berners-Lee and D. Connoly. "Delta: An Ontology for the Distribution of Differences Between RDF Graphs", 2004. http://www.w3.org/DesignIssues/Diff.
2. J. Heflin, J. Hendler, and S. Luke. "Coping with Changing Ontologies in a Distributed Environment". In *AAAI-99 Workshop on Ontology Management*, 1999.
3. M. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. "Ontology versioning and change detection on the web". In *Procs of EKAW'02*, 2002.
4. A. Mallea, M. Arenas, A. Hogan, and A. Polleres. On blank nodes. In *Procs of the 10th Intern. Semantic Web Conference (ISWC 2011)*. Springer, October 2011.
5. J. Munkres. Algorithms for the assignment and transportation problems. *J-SIAM*, 5(1), 1957.
6. B. Schandl. Replication and versioning of partial rdf graphs. ESWC'10, 2010.
7. Y. Tzitzikas, C. Lantzaki, and D. Zeginis. "Blank Node Matching and RDF/S Comparison Functions". ISWC'12, 2012.
8. D. Zeginis, Y. Tzitzikas, and V. Christophides. "On the Foundations of Computing Deltas Between RDF Models". In *Procs of ISWC-07*, 2007.

[4] http://semweb.salzburgresearch.at/apps/rdf-gravity