Learning to Detect Abnormal Semantic Web Data

Yang Yu
Lehigh University
Bethlehem, PA, 18015, USA
yay208@cse.lehigh.edu

Xingjian Zhang
Lehigh University
Bethlehem, PA, 18015, USA
xiz307@cse.lehigh.edu

Jeff Heflin
Lehigh University
Bethlehem, PA, 18015, USA
heflin@cse.lehigh.edu

Categories and Subject Descriptors

I.2.6 [**Learning**]: Knowledge acquisition

General Terms

Algorithms, Experimentation, Performance

Keywords

detecting abnormal data, Semantic Web, learning

1. INTRODUCTION

Numerous problems could happen in the generation process for Semantic Web data that is usually gathered from heterogeneous sources by using a variety of tools [3]. Recently some works [1, 2, 3, 4] began to focus on the quality of Semantic Web data. However since the Semantic Web represents many points of view, there is no objective measure of correctness for all Semantic Web data. Therefore, we consider using an abnormality heuristic that could indicate a data quality problem at the triple level. We recognize that not all abnormal data is incorrect (in fact, in some scenarios the abnormal data may be the most interesting data) and thus leave it up to the application to determine how to use the heuristic. The essential idea of this work is based on the fact that a statement can get supporting evidence if it can be entailed from other data. Consider the statement A advises B: in some situations where this is true, there are also statements such as A is the principal investigator of project C, B works in C. This rule is clearly not certain. Yet, when combined with other forms of evidence, it can provide support for the advises relation.

To detect incorrect data, ideally we can directly learn characteristics of them. But incorrect data have too many forms. So we check if the data lacks sufficient normal patterns compared to the majority of the data. Still using the advises relation example above, we change the first statement into B advises A (assuming advises is not subPropertyOf advises$^-$). Then our predictability on this statement would be low, because the context is inconsistent with a probabilistic rule existing in many other contexts. Although this probabilistic rule does not always hold, various rules in context can collaboratively give certain support. Note that there are many possible arbitrary relations that can be used to describe any two objects on the Semantic Web, but the notion of *significant relation* used in this work is tied to the ontologies used by the system.

K-CAP'11, June 26–29, 2011, Banff, Alberta, Canada.
ACM 978-1-4503-0396-5/11/06.

2. CONTEXT BASED IDENTIFYING SIGNIFICANT RELATION

Formally, our problem is defined as: given the pair u of subject s and object o, how significant is some relation p between the pair u (written $y_{u,p}$). The pair u_k of subject s_k and object o_k is another pair (s_k != s or o_k != o) having the semantic relation p. The $y_{u,p}$ can be measured by the overall similarity between the pair u and all the pairs u_k (equation 1), where U_p is the set of all pairs that have the relation p and sim() is the similarity function on two contexts which will be introduced in Section 3.3.

$$y_{u,p} = \frac{1}{|U_p|} \sum_{u_k \in U_p, u_k != u} sim(u_k, u) \quad (1)$$

We define the semantic connection as $\langle r_1, r_2, ..., r_n \rangle$, where r_i is a relation. Then the context for a pair u is defined over a semantic connection space which is a vector space consisting of all possible semantic connections (the first part in following equation), where n_{u,c_i} means the number of instantiations of the semantic connection c_i ($i \leqslant m$) between the pair u. To get more supporting evidence for a predicate usage between two instances, for each instance, we build a set of similar instances including itself and call this set the expanded set. Because the semantic connections between the two expanded sets are partially similar to the semantic connections between original pair u, they are treated as partial semantic connections between the original pair. The full contetxt of pair u are represented below.

$$\begin{aligned} V_u &= [n_{u,c_1}, n_{u,c_2}, ..., n_{u,c_m}] + \alpha[n_{\bar{u},c_1}, n_{\bar{u},c_2}, ..., n_{\bar{u},c_m}] \\ &= [n_{u,c_1} + \alpha n_{\bar{u},c_1}, n_{u,c_2} + \alpha n_{\bar{u},c_2}, ..., n_{u,c_m} + \alpha n_{\bar{u},c_m}] \end{aligned}$$

In similarity measuring, the partial matching between different connections should affect the similarity between vectors. Considering that, we define the similarity between vectors as the sum of the similarities between all pairs of connections divided by the multiplication of the magnitude of two vectors (equation 2).

$$sim(u', u) = \frac{1}{||u||||u'||} \sum_{i=1}^{m} \sum_{j=1}^{m} n_{u,c_i} n_{u',c_j} s(c_i, c_j) \quad (2)$$

$$s(c_i, c_j) = s(< r_{i1}, r_{i2}, ..., r_{in} >, < r_{j1}, r_{j2}, ..., r_{jn} >)$$
$$= \prod_{k=1}^{n} x_{ik,jk} \quad (3)$$

where $x_{ik,jk}$ is the similarity between property r_{ik} and r_{jk}.

3. LEARNING PREDICATE SIMILARITY

The model we used is modified from its application in the tag prediction problem [5]. For a pair u of subject and object, the algorithm ranks predicates by $y_{u,p}$. The objective function (equation 4) maximizes the ranking statistic AUC (area under the ROC-curve).

$$AUC(\hat{\theta}, u) = \frac{1}{|P_u^+||P_u^-|} \sum_{p^+ \in P_u^+} \sum_{p^- \in P_u^-} h(y_{u,p^+} - y_{u,p^-}) \quad (4)$$

The h(x) is a continuous sigmoid function

$$h(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Then using gradient descent, AUC has to be differentiated with respect to all model parameters and for each pair u $\in P_s$, P_u^+ is the set of predicates that are already used between the pair u while P_u^- is the set of predicates that are not used between the pair u. The overall optimization task with respect to the ranking statistic AUC and the observed data is then:

$$\arg\max_{\hat{\theta}} \sum_{u \in Ps} AUC(\hat{\theta}, u) \quad (6)$$

The model parameters $\mathbf{x}$ which is a vector of all possible pairs of predicate similarity introduced in Section 3.3 are updated $\frac{\partial}{\partial \mathbf{x}} AUC(\hat{\theta}, u)$. We note that this equation contains a lot of computations that can be reused for each round, e.g. the derivative of the similarity between two connections are not changed within each iteration. So we use some memoization techniques to save huge amount of repeated computations. After each iteration, update the memoized table once. Thus for each pair u, the $\mathbf{x}$ are updated as follows:

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \gamma \cdot \frac{\partial AUC}{\partial \mathbf{x}} \quad (7)$$

where γ is the learning rate which we have set as 0.05. This equation means after the model learns from each observed triple to increase the gap between the positives and the negatives, it updates the model parameters, i.e. predicate similarities, based on the learning rate.

4. EXPERIMENTS

After removal of test triples, the experiment process generally is as follows. First, the system randomly selects some training samples and builds the contexts for them. Second, the system learns model parameters on training samples, given their initial values. Finally, we input test samples with unknown predicate to the system, both positive and negative, and check the result of entailed predicate with highest score. For positive samples, the system is expected to entail the correct predicate, which means the system can detect the abnormality if the predicate is incorrect. For negative samples, it is expected that no relation between the objects entailed by the system is above a certain threshold β and then the system report it as no relation. Thus all experiments use precision, recall and F-measure.

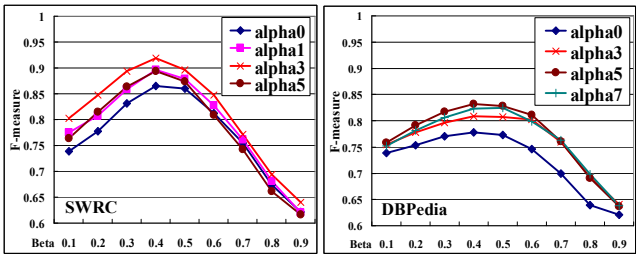

Figure 1: The effect of different expansion factor α and different credible relation threshold β.

The experiment compares the performance when the expansion factor α (Section 3.2) and the threshold β (Section 5.1) varies (shown in Figure 1). From the results, we see that the system without expansion ($\alpha = 0$) is worse than any systems with expansion and among those the systems with $\alpha 3$ ($\alpha = 0.3$) and $\alpha 5$ ($\alpha = 0.5$) are the best on two data sets. To not overwhelm readers, the lines with other alpha values are not shown here. The reason DBPedia needs more context expansion is that it has less relational descriptions for instances than SWRC. For β, the system performs the best on both data sets when it is 0.4.

5. CONCLUSION

The essential idea of this work is to use probabilistic rules in the context of a triple and the context of typical triples to generate a measure of abnormality. The probabilistic rules are learned from semantic connections between objects in triples. To deal with the open world assumption underlying the Semantic Web data, the system uses three mechanisms, i.e. enriching the context, a novel context comparison mechanism and a learning model considering the missing triples. The approach is mainly based on data itself without ontological inference and unsupervised learns from a set of data sources that are generally correct.

6. REFERENCES

[1] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the ACL*, 2002.

[2] C. Fürber and M. Hepp. Using semantic web resources for data quality management. In P. Cimiano and H. S. Pinto, editors, *EKAW*, volume 6317 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2010.

[3] Y. Lei, V. Uren, and E. Motta. A framework for evaluating semantic metadata. In *Proceedings of the 4th international conference on Knowledge capture*, K-CAP '07, pages 135–142, New York, NY, USA, 2007. ACM.

[4] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on ąśEvaluation of Ontologies for the Web*, 2006.

[5] S. Rendle, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD'09*, pages 727–736, 2009.