Type Inference on Noisy Linked Data

Jing Chen (2819**), Luming Pan (2819****)**

Submission date: 2017-05-29

Southeast University-Monash University Joint Graduate School, Suzhou, China

Abstract

Type information of entities which can be utilized to do semantic search is valuable in linked data. However, many linked data bases are incomplete in type information. Traditional research of type inference method can find missing types by means of logic-based reasoning method, but it may become invalid in those data with incomplete or incorrect schema. In this paper, we propose a text-classification procedure in six steps to predict missing type information in linked data. Our method can be a supplement of previous text-based and linked-based inference methods.

Keywords: Type inference, linked data graph, weighted BOW

1. Introduction

A large number of linked data bases are built to publish structured and related data for web semantic retrieval. However, a common problem happens in the linked data base is that an entity loses its type information, because linked data base is constructed in the crowdsourcing way and human may make mistakes during the constructing period (Acosta et al., 2013). In sematic search area, entity type on linked data is an important element for sematic query. Therefore, type inference, which is also called type prediction, has become an essential work to improve the quality of linked data base for better sematic search. During our research, we found traditional approach to realize type inference is the logic-based reasoning method. However, such logic-based method has two main limitations. First, it should expend effort to process the noisy problem on linked data and the improvement of predict accuracy is not obvious (Ji et al., 2011). Second, it cannot be reasoning across data sets (Paulheim and Bizer, 2013). Gangemi et al. (2012) proposed a tool named *Tipalo* based on text and link information to predict the missing type rather than using the logic-based reasoning method. This method can improve the predict result accuracy and avoid the noisy problem in linked data, but a limitation of this method is that it is only suitable for the database *DBpedia*. Because this method utilizes the abstract information from Wikipedia that links out of the data base and conducts semantic analysis on the content of the text to predict the type. When the predicted entity doesn't have abstract text in Wikipedia, this *Tipalo* tool cannot predict the type.

Therefore, the main problem of our research problem is to find a type inference method and meet two requirements. First, the method can be applied to different linked databases universally. Second, the method can ensure a relatively high prediction accuracy.

2. Objectives

The aim of our research is to utilize the text information which has already existed in the linked data base to predict the missing type of entities. The main contribution of this paper is that we suppose to use a weighted bag-of-words (BOW) model to handle and represent the isolate text information of a target entity and its adjacent entity or links. This method can be applied to different data bases rather than a database. The *Tipalo* tool uses the abstract information that links to Wikipedia, but our approach only cares about the text information that is already included in the linked data base. What's more, we classify the relations on linked data to ensure and improve the predict result accuracy. Such relation classification method is inspired by *SD-type*, which assumes that different relation has different indications of entity type (Paulheim and Bizer, 2013).

The detail of our procedure to predict missing type will be explained in section 3.

3. Methodology

The architecture of our type inference approach has six steps: parsing the linked data, classifying each object's linking relation, representing each object's texture information by bag-of-words model, constructing a weighted-BOW for each object, estimating weight of weighted BOW in the training set, using K-nearest Neighbors algorithm in testing set, and evaluation.

Firstly, linked data graph is usually presented as various triples of Resource Description Framework(RDF). A triple is like subject-predicate-object, subject and object are objects on the linked data, and predicate is the relation linking subject and object (Bizer et al., 2009). A predicted object and its adjacent objects are linked with a specific relation. PDF parser can use many existing tools such as *jena* implemented by java. Moreover, it is not difficult to write one since the parsing principles are very simple.

Secondly, linking relations of each predicted object are classified into 4 categories based on 2 principals. The first principal is based on direction of relation towards a predict object. There are two directions, which are incoming and outgoing. Direction of relation is from subject to object. If a predicted object is the subject of a relation, then the relation is an outgoing relation for the object. Vice versa. The second principal is based on the domain and range of the relation. The domain of relation is type of subject and the range of relation is the type of object. If the domain and range of relation are same, then the relation is a homogeneous relation. If the domain and range of relation are different, then the relation is a heterogeneous relation. Based on these two principals, relations can be classified into 4 class: incoming-homogeneous relation, incoming-heterogeneous relation, outgoing-homogeneous relation and outgoing-heterogeneous relation.

Thirdly, we obtain texture information of each object and apply natural language processing (NLP) method on it. Because linked data is usually extracted from semi-structured web page, the texture information is rambling. For example, a relation indicating the place of birth is presented as “birthPlace”. It should be processed to “birth place”. Therefore, BOW model can be used in the texture information. BOW is a simplified text representation model, in which document is regarded as the frequency distribution of words without considering grammar and the order of words.

Fourthly, we construct a weighted-BOW consisting of texture information of object and its adjacent objects for each object. A weighted-BOW of an object will be presented like:

$$Weighted\_BOW(o) = w_{in-ho} \cdot BOW(o_{in-ho}) + w_{in-he} \cdot BOW(o_{in-he}) + w_{out-ho} \cdot BOW(o_{out-ho}) + w_{out-he} \cdot BOW(o_{out-he}) + w_o \cdot BOW(o). \quad (1)$$

$BOW(o_{in-ho})$ means the BOW of o’s adjacent object with incoming-homogeneous relation. In addition, w_{in-ho} is used to balance the weight of it. Similarly, in-he means incoming-homogeneous relation, out-ho means outgoing-homogeneous relation, and out-hete means outgoing-heterogeneous relation. BOW(o) means itself, which includes texture information of its as well as its relations’. How to estimate the weights of weighed-BOW will be discussed in the fifth step.

Fifthly, we estimate the weights of weighted-BOW. In this case, the weighs are shared overall linked data, which means there are only five weights on the linked data. Since the weight of a BOW indicates the importance of it, we consider the weight means the prediction ability of the BOW. Therefore, a simple weight estimating approach is like this:

$$W_x = \frac{Precision_x(t)}{\sum_i Precision_i(t)}, i, x \epsilon \{in - ho, in - he, out - ho, out - he, o\} \quad (2)$$

$Precision_x(t)$ means the precision of BOW of adjacent object with x relation predicting the type named t. Before weight estimation, the linked data should be divided into training set and test set. Training set is used to do weight estimation, since it consists of objects with correct type. Moreover, the classification and prediction can use Naïve Bayes or K-nearest Neighbors classifier. We decide to use K-nearest

Neighbors.

Sixthly, we obtain model and training set as well as test set in the fifth step. Thus, we apply the model from fifth step to test set, and then we get the evaluation of our method. There is also another evaluation approach from an evaluation framework of knowledge base (Neelakantan and Chang, 2015). It constructs training set from previous version of linked data and consider the newest version of linked data is the predicted facts. By matching the result of training set and predicted facts, the evaluation is easy to obtain.

4. Novelty

According to our research, in linked data entity type inference area, most studies focuses on expending the relation of entity or spare effort to deal with the inconsistent noisy problem in the logic-based reasoning type inference method. Resent years several researchers start to utilize the text information in linked data to predict the missing type. But as the introduction part explained, there are some limitation of three relevant papers. Unlike the traditional logic-based and link-based type (Gangemi et al., 2012) inference methods, our approach utilizes the entity and class objects' own text information in the linked data base.

Our method could be a supplement of previous text-based or linked-based inference methods and can give an evidence that the text information in linked data itself can be used to predict the missing type.

5. Conclusion

Type missing problem is common on linked data bases and can influence the sematic search result. Type inference can improve the quality of linked data base. Our project gives out a procedure to predict type utilizing text information on linked data and use classification procedure to predict type (Sleeman et al., 2015), which can make up for traditional limitations. The procedure consists of six steps, which includes parsing linked data graph, classifying relations, NLP, weighted BOW representation, estimating weights and evaluation. We propose to use a weighted BOW model to solve the representation of isolate text problem for further classification and use the relation classification approach to improve the prediction result. Moreover, two means of evaluation will be used to assess the performance of our method.

References

Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., & Lehmann, J. (2013). Crowdsourcing linked data quality assessment. *International Semantic Web*

Conference, 260-276.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, 205-227.

Gangemi, A., Nuzzolese, A., Presutti, V., Draicchio, F., Musetti, A., & Ciancarini, P. (2012). Automatic typing of DBpedia entities. *The Semantic Web–ISWC 2012*, 65-81.

Ji, Q., Gao, Z., & Huang, Z. (2011). Reasoning with noisy semantic data. *Extended Semantic Web Conference,* 497-502.

Neelakantan, A., & Chang, M. W. (2015). Inferring missing entity type instances for knowledge base completion: New dataset and methods. *arXiv preprint arXiv:1504.06658*.

Paulheim, H., & Bizer, C. (2013). Type inference on noisy RDF data. *International Semantic Web Conference,* 510-525.

Sleeman, J., Finin, T., & Joshi, A. (2015). Entity type recognition for heterogeneous s emantic graphs. *AI Magazine,* 36(1), 75-86.