A Literature Review on
Type Inference on Noisy Linked Data

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

Submission date: 2017-05-12

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

Abstract

This report focuses on reviewing different approaches to do object type inference on noisy linked data. Introduction part gives the motivation of this review and briefly introduces the background information of our research topic. Firstly, a traditional logic-based method named reasoning and noisy data problems in this method are reviewed. Then we review three type inference approaches based on linked data features: the linked-based approach, the natural language processing approach, and the classification approach. All these approaches can avoid or tolerate the noise data problem. Finally, we assess these type inference methods and propose an assumption which combines three approaches we review.

Keywords: Type inference, Linked data, Entity type

1. Introduction

Linked data is a graph model and aims to expose and connect related data from diverse sources on the Semantic Web. Tim Berners-Lee outlined four 'Linked data rules' for publishing linked data on the Semantic Web [1]. According to the rules, the linked data should use Uniform Resource Identifier (URI) as names for things and the URIs should use the Resource Description Framework (RDF) standards to provide useful information [1]. Type information of objects in linked data plays an important role in knowledge bases, because a certain object type can be one of the atomic building blocks of knowledge bases. However, most linked data is constructed in a crowdsourcing way relying on basic human abilities, which can lead to incorrectly or missing object type

[2]. Hence, it is needed to find an approach to predict missing types and improve the quality of linked data. The research of type inference usually relies on logical inference to get missing types, but this approach is likely to multiply errors in noisy linked data.

The main motivation of this review is to provide an overview of current approaches of type inference in linked data. Besides, we analyze the advantages and limitations of the type inference methods in each paper. In the following parts, section 2 introduces the scope and method of this review. Section 3 reviews four papers which related to noisy linked data and different type inference approaches. At last, in section 4, we summarize this review paper and propose an assumption.

2. Scope and method

The main purpose of our research is to do object type inference on linked data. So, we focus on reviewing different methods that can predict the type information on linked data. Firstly, we review articles about reasoning, which is a traditional type inference method, to understand the structure of linked data and find out problems in reasoning approach. Then, we review three type inferences methods based on linked data features that can avoid the problems in traditional logic-based approach, because we want to combine different methods and find a new approach to predict the type information of objects in linked data.

To conduct this review, we firstly asked our supervisor for his suggestion to attest the scope we review and all the research papers we find can have strong correlation with our research topic. All the papers we selected are from the most important publications or conference proceedings in Semantic Web area. Then, we use Monash University Library Database and Google scholar to search for all the papers. After extensive reading, we finally choose six papers to read in detail to review their approaches.

3. Body of the review

3.1 Problems in traditional type inference approach

The traditional approach to predict object type information in the Semantic Web is the use of logic-based reasoning. We review two articles in this part, and find the main problem as well as disadvantages in the traditional reasoning approach.

In the first article, Ji et al. [3] point out that noise in the linked data can be a problem influencing on the accuracy of predicted result. The authors explain that the noise of linked data caused by atypical using of RDF vocabularies, or use of undefined classes

and properties. Some linked data are transformed from semi-structure or even unstructured data using Nature Language Processing (NLP) technologies. Ontologies inconsistency in a big noise data problem in traditional type inference approach [3]. In this article, the authors also point out that inconsistent linked data can be pre-processed with those triples causing inconsistencies dropped according to some heuristic measures.

In the second article, Paulheim and Bizer [4] conducted an experiment with DBpedia knowledge base to illustrate the noise data problems that can occur with traditional reasoning approach. The authors conclude that noise data problems exist because traditional reasoning is only useful under two situations: one is when both the knowledge base and the schema do not contain any errors and another is when the schema is only used in ways foreseen by its creator.

However, both assumptions of these situations are not realistic for large and open knowledge bases. The authors point out that although reasoning seems the straight forward approach to solve the problem of completing missing types, it is not applicable for large, open knowledge bases [4].

3.2 Type inference in link-based approach

The first kind of type inference approach based on linked data feature rather than logic-based reasoning we review is an object link-based method. Paulheim and Bizer [4] propose a type inference approach based on object links, named *SD-Type*, which is based on link and statistical theory. The following is the basic method of *SD-Type*.

On linked data, link between two instances is named relation or predict. Moreover, a relation between two instances is expressed as a statement like structure: *s p o*, in which *s* means subject, *p* means predict and *o* means object. In this article, the *SD-Type* approach makes an assumption: an instance, belongs to a particular type, if it has a certain relation with some objects. In addition, since a certain instance will have many relations with other instances, there should be different indicators for different links to vote for the certain instance's type. Probability of an instance belonging to a type will be calculated by weighted sum of its adjacent objects' probability based on their particular links. There are four steps in *SD-Type* approach [4]: 1. Inputting data; 2. Computing basic distribution; 3. Computing weights and conditional probabilities; 4. Materializing missing type. These steps can be used in our research to define and access the relation between the objects to predict object type.

In this article, *SD-Type* can address type prediction on noisy linked data problem, because it mainly uses link's feature and links are normally correct and complete. *SD-Type* has been applied to practice and work well. On the one hand, it still has some limitations: it cannot do type inference across dataset, since it only can calculate probability distribution in one dataset; it does not utilize more information on linked data. On the other hand, it is a classic method and inspiring for our research, since it takes the differences of relations with adjacent instances into account. Our research can use text information of adjacent instances, and try to give them different weight based on their relation type. The weights will be estimated in different way.

3.3 Type inference in classification approach

The second type inference approach based on linked data features we review uses the classification approach. Sleeman et al. [5] propose an approach by finding attributes of objects on linked data to define types and then predict. A dictionary for each type was built by calculating the information gain for each attribute of object. The authors propose that objects without type information can be labeled with a type through supervised classification.

They use a distance mapper to measure the similarity of predicate labels on linked data. And then, the authors use a Support Vector Machine (SVM) to develop a model for each object type. They use a training set of labeled data and map the objects in the labeled data to dictionaries that built by the authors and use a set of mappers which each emit a score that is used as a feature of the objects missing types. The features are later used with a supervised algorithm to create a classification model. Then, the author performs mappings on unlabeled data and classify these objects using the supervised model. The results of this classification are instances classified as different types such as person, location, or organization. This approach enables to map attributes from different domains, hence supporting heterogeneous data and avoid the problem of noise data in the traditional type inference method.

In this article, every predicate object on linked data is mapped to an entity type and each entity type is ontologically defined giving way to hierarchical relationships, equivalent relations, and pattern similarity relationships [5]. After the review, we think we can imitate this method to take an unfiltered approach to use the ontological definitions to then find candidate types and predict unknown types. However, their work differs in that they focus merely on attributes as type indicator. In our research, we can consider multiple indicators covering comprehensive linguistic information on linked data, and try to use different strategies to figure out the type information of an object.

3.4 Type inference based on natural language information

The third approach of object type inference is based on natural language information and refer to other knowledge bases. Gangemi et al. [6] propose a type inference model, named *Tìpalo*, which is particularly used in DBpedia database. DBpedia extracts structured information from Wikipedia and links other resource sets to Wikipedia by crowd-sourced work. *Tìpalo* uses this characteristic of DBpedia. It identifies an instance's type by its texture description from its corresponding Wikipedia pages.

In this article, there are five steps for typing instances on DBpedia based on the authors' texture definition: Firstly, instance's definition is the most important and shortest sentence extracted from comparable Wikipedia pages by analyzing characteristic of sentences in Wikipedia web pages; Secondly, the sentence should be deeply parsed to obtain a logical expression like Web Ontology Language (OWL) graph by an online tool called *fred*; Thirdly, *fred* output graph is parsed by a set of graph patterns (GP), to recognize whether an object of the graph is an instance or a class, and classes detected in this step will be used as candidate types for the fifth step; Fourthly, Word Sense Disambiguation (WSD) done by a WSD tool is necessary, since it uses words from an unknown, raw and long paragraph; Fifthly, the disambiguated instance can easily achieve its Wikipedia entity type by type matching with other like *WordNet*.

Tìpalo can handle with type inference on noisy linked data, for it utilizes text information from Wikipedia without worrying about noise on DBpedia. It has been proven to work well on DBpedia. However, methods used in *Tìpalo* requires dataset like DBpedia, which should have or link to abstract data related to the instance. It relies on the text data of specific structure, and cannot be applied to other kinds of linked data set. So, in our research, we can also make use of texture information on linked data. However, it can work on any kind of linked data, since we use all of texture information inside database, not asking for a specific instance description.

4. Conclusion

In this review, we focus on reviewing approaches of objects type inference on linked data which can avoid noisy data problem. We review the traditional logic-based method for type inference and find the noise data problems faced in traditional method. After the review, we find that there are ample links and linguistic information on linked data that can be used to predict the object type. The influence of noise data can be reduced in this way. We propose to combine the merits of the methods reviewed in Section 3.2, 3.3, and 3.4 to solve the type inference problem and get a better type predicted result.

(Totally 2,107 words)

References

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

[2] Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., & Lehmann, J. (2013). Crowdsourcing linked data quality assessment. *International Semantic Web Conference* (pp. 260-276). Springer Berlin Heidelberg.

[3] Ji, Q., Gao, Z., & Huang, Z. (2011). Reasoning with noisy semantic data. *Extended Semantic Web Conference* (pp. 497-502). Springer Berlin Heidelberg.

[4] Paulheim, H., & Bizer, C. (2013). Type inference on noisy RDF data. *International Semantic Web Conference* (pp. 510-525). Springer Berlin Heidelberg.

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

[6] 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.