Building Chinese Linking Open Data

FANG Yishu (246[redacted]) and XIA Yingying (246[redacted])
Last updated: 31 May, 2013

Abstract:

Knowledge base plays a critical role in the successful construction of the Semantic Web (SW). Linking Open Data (LOD) is considered as an ideal tool to describe the knowledge base of the SW. However, native Chinese LOD (CLOD) rarely exists and Chinese information also rarely appears in existing multilingual LOD datasets. In this paper, we build a prototype dataset of CLOD and focus on the solutions to two representative problems: how to discover equal relationships of different instances extracted from heterogeneous websites accurately and comprehensively, and how to effectively merge small ontologies built from different sites to form a large one. To solve these problems, new approaches are given to discovering equal relations and ontology merging in heterogeneous Chinese websites context.

Keywords: Semantic Web, Linking Open Data, Chinese Linking Open Data, Ontology Merging, Web of Data

1. Introduction

Knowledge base plays a critical role in the successful construction of the Semantic Web (SW). With the development of the SW, hundreds of datasets have been published under Linking Open Data (LOD) community project (Berners-Lee & O'Hara, 2013). As LOD tries to connect the distributed data across the entire Web, it is an ideal tool to describe the knowledge base of the SW. However native Chinese LOD (CLOD) rarely exists and Chinese information also rarely appears in existing multilingual LOD datasets. Regarding LOD datasets built directly on Chinese information, to our knowledge, only Zhao (2010) and Niu et al. (2011) are trying to build native CLOD datasets. However, they are only

concentrated on medicine and encyclopedia information, respectively. While inspecting existing multilingual datasets, such as freebase and DBpedia, they are all constructed using English and attached with Chinese translation. Chinese information within those datasets is incomplete and lacks real-world usage within Chinese language context (Niu et al., 2011). With these two reasons, we are trying to build a native CLOD dataset from a wide range of heterogeneous Chinese websites.

2. Objectives

Our main objective in this project is building a prototype dataset of CLOD. That dataset should contain comprehensive Chinese information and have real-world usage.

While trying to build the CLOD dataset, there are many challenging problems we have to take into consideration. We focus on dealing with two representative problems. The first one is how to discover equal relationships of different instances extracted from heterogeneous websites accurately and comprehensively. The second problem is how to effectively merge small ontologies built from different sites to form a large one.

3. Methodology

In our project, four issues have to be addressed: what kind of data should be extracted from existing sites, how to represent those data, how to discover synonyms from those data and how to merge small ontologies to form a large one based on those synonyms.

3.1 Category Extraction

The first step of building the CLOD dataset is extracting information from existing Chinese web sites of various types. The information hierarchy based on which the site is built on is exactly we want to obtain. Taking Amazon as an example (the red section will be level 1 information, and the blue section will be level 2 information under book section, by clicking “Textbooks”, there will be more categories which are not shown here):

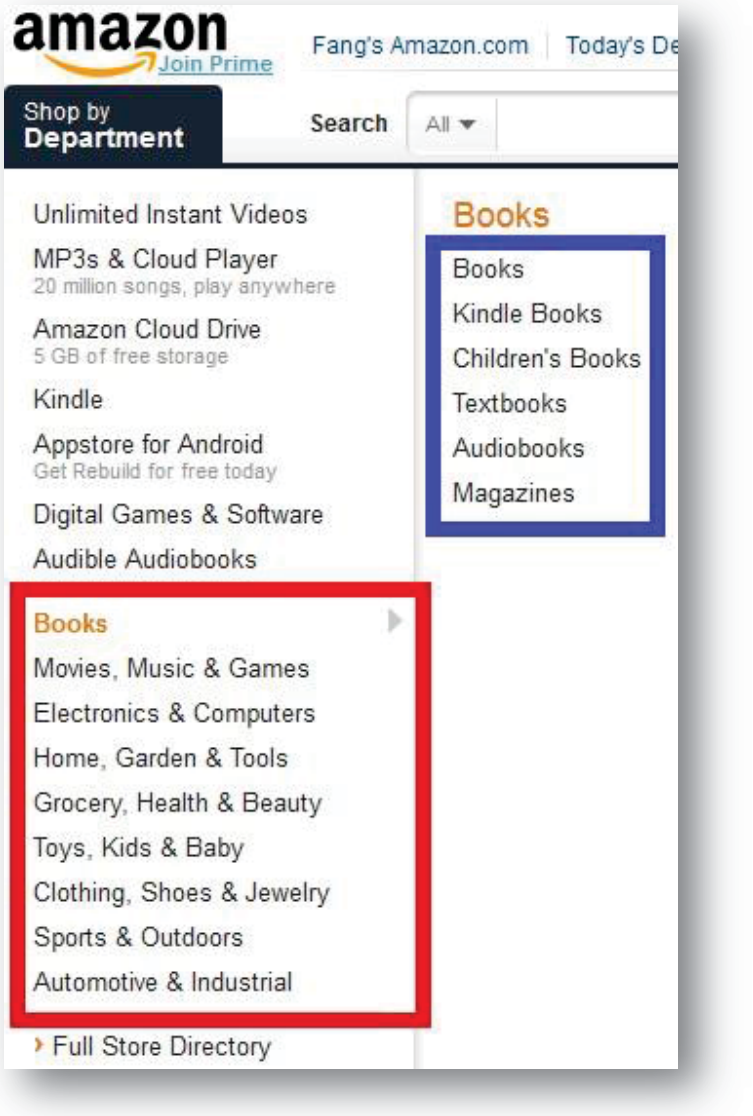

Figure 1: screenshot of Amazon.com

The hierarchical information we want to get is something like this and this will be one of our small ontologies:

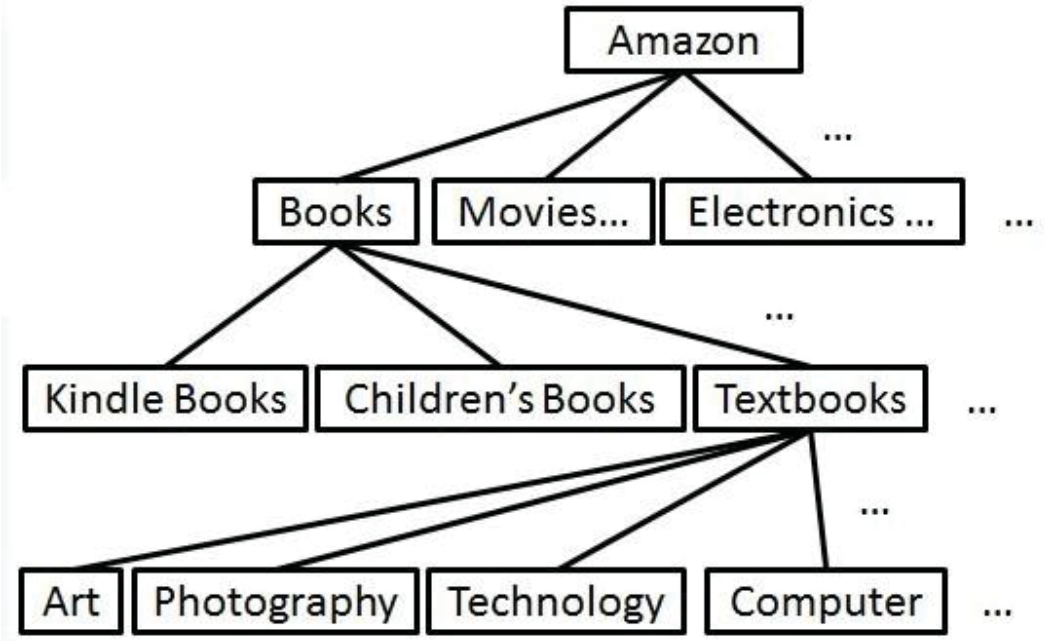

Figure 2: Ontology in a tree style of Amazom.com

We have extracted 711381 categories from 50 Chinese popular web sites. Figure 2 shows the category tree corresponding to Amazon.com. In our practical work, each site has a category tree corresponding to it.

3.2 Category Representation

We use three different mechanisms to represent a category:

1) Name of the category,
2) Related category set of the category,
3) Related category vector of the category.

The naive way to represent a category is using its category name. However, name is not enough to fully describe that category. For example, "NYC" and "New York City" are synonyms, but these two words have significant difference in string representation which may lead to the incorrect judgment. Inspired by work of Shen et al. (2007), we consider two other representation methods, i.e., representing a category by its related category set and representing a category by its related category vector. These methods can help us to measure the similarity of categories.

To construct the related category set, we submit each category as a query word to BaiduBaike and BaiduZhidao, and then we can get the open categories from Baidubaike and the question categories from Baiduzhidao. These two kinds of categories form the related categories of the query word and each category has several related categories. The related categories can be used to define the related category set RCS(c) of a category c. We can also use them to define the related category vector, denoted as a vector: $(rc_1, rc_2 \ldots rc_n)$, where $\mathrm{rc}_i (i = 1,2 \ldots n)$ is the number of times i^{th} related category occurs. According to the related category-based category representation, if two categories have similar related categories, these two categories are probably synonyms, though they are significantly different in their category name.

3.3 Synonym Discovery

The synonym discovery mechanisms we use in this paper are based on three most widely used similarity measures: Levenshtein distance (Schimke et al., 2004; Yujian & Bo, 2007), Jaccard similarity and cosine similarity (Strehl et al., 2000). These three similarity measures are applied to three category representation methods, respectively.

1) **Similarity based on category name.** This method is actually string matching based on the Levenshtein distance. The similarity between categories c_1 and c_2 based on category label is defined as:

$$\mathrm{BOWISim}(c_1, c_2) = \frac{Ld(c_1, c_2)}{max \text{ Error! Bookmark not defined.}}$$

Where $|c|$ the string length of category name of c, and $\mathrm{Ld}(c_1, c_2)$ is the Levenshtein distance between c_1 and c_2.

2) **Similarity based on related category set.** According to the related category-based category representation, each category can be regarded as a set of related categories. So we can give the definition of similarity function as follows:

$$\mathrm{BORCSSim}(c_1, c_2) = \frac{|RCS(c_1) \cap RCS(c_2)|}{|RCS(c_1) \cup RCS(c_2)|}$$

Where $BCS(c)$ is the related category set of category c. Actually this equation is the Jaccard similarity between sets $RCS(c_1)$ and $RCS(c_2)$.

3) **Similarity based on related category vector.** Same as the second point, the similarity based on related category vector also results from the related category-based category representation. The similarity is defined as :

$$\mathrm{BORCVSim}(c_1, c_2) = \frac{\sum_{rc \in RCS(c_1) \cap RCS(c_2)} rc(c_1) \cdot rc(c_2)}{\sqrt{\sum_{rc \in RCS(c_1)} rc(c_1)^2 \sum_{rc \in RCS(c_2)} rc(c_2)^2}}$$

Where rc(c) denotes the number of times the related category rc of category c occurs. As shown above, this equation is the cosine similarity between vectors c_1 and c_2.

Finally, using logistic regression to do a fit of those above three algorithms, we get:

$$\mathrm{NODESim}(c_1, c_2) = F(BOWISim(c_1, c_2), BORCSSim(c_1, c_2), BORCVSim(c_1, c_2))$$

The output will be either true or false denoting whether c_1 and c_2 are synonyms.

3.4 Ontology Merging

With the above works done, we can use those synonyms found to merge those ontologies (i.e. discover `owl:sameAs` relationship), but only synonyms are not enough. For example, consider the word "apple"; it may refer to both a kind of fruit and the Apple Company. In that case, context information must also be taken into consideration.

If we take all sub-categories of c_1 (denoted by sub-categories(c_1)) and all sub-categories of c_2 (denoted by sub-categories(c_2)) as the context of c_1 and c_2, respectively, then we can define a more suitable similarity measure. The idea is that, if there are enough similar pairs between sub-categories(c_1) and sub-categories(c_2), then c_1 equals c_2. We define a new similarity measure as follows:

$$\mathrm{TMSim}(c_1, c_2) = \frac{\delta}{\min\{Nodes(c_1), Nodes(c_2)\}}$$

Where Nodes (c) is the number of all nodes of the tree which takes c as the root, δ is the number of synonyms pairs between sub-categories(c_1) and sub-categories(c_2). When that value reaches a threshold, we will determine the `owl:sameAs` relationship between c_1 and c_2.

4. Novelty

Using the methods described above, we have successfully built the first comprehensive dataset of CLOD. That dataset is complete and have real-word usage unlike Chinese portion of DBpedia and Freebase. Besides, this dataset have covered more information than previous CLOD attempting by Zhao (2010) and Niu et al. (2011).

5. Conclusion and Significance

In this paper, we present the rational base and process of the building a CLOD dataset. We mainly focus on two representative problems: how to discover equal relationships of different instances extracted from heterogeneous websites accurately and comprehensively and how to effectively merge small ontologies built from different sites to form a large one. The first problem is solved by the introduction of three similarity algorithms and the second one is achieve by taking context information into consideration.

This prototype dataset we build is the first comprehensive CLOD dataset with real-world usage. Based on that, many meaningful researches can be conducted, such as semantic website recommendation, Chinese search engine optimization based on CLOD and so on.

References

Berners-Lee, T., & O'Hara, K. (2013). The read–write Linked Data Web. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1471-2962.

Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., & Yu, Y. (2011). *Zhishi. me-weaving Chinese linking open data.* Paper presented at the The Semantic Web–ISWC 2011, 205-220.

Schimke, S., Vielhauer, C., & Dittmann, J. (2004). *Using adapted levenshtein distance for on-line signature authentication.* Paper presented at the Proceedings of the 17th International Conference on Pattern Recognition, 931-934.

Shen, D., Qin, M., Chen, W., Yang, Q., & Chen, Z. (2007). *Mining web query hierarchies from clickthrough data.* Paper presented at the Proceedings of the National Conference on Artificial Intelligence, 341-346

Strehl, A., Ghosh, J., & Mooney, R. (2000). *Impact of similarity measures on web-page clustering.* Paper presented at the Workshop on Artificial Intelligence for Web Search (AAAI 2000), 58-64.

Yujian, L., & Bo, L. (2007). A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(6), 1091-1095.

Zhao, J. (2010). Publishing Chinese medicine knowledge as Linked Data on the Web. *Chinese medicine, 5*(1), 1-12.