On Publishing Chinese Linked Open Schema

Haofen Wang[1], Tianxing Wu[2], Guilin Qi[2], and Tong Ruan[1]

[1] East China University of Science and Technology, Shanghai, 200237, China
{whfcarter,ruantong}@ecust.edu.cn
[2] Southeast University, China
{wutianxing,gqi}@seu.edu.cn

Abstract. Linking Open Data (LOD) is the largest community effort for semantic data publishing which converts the Web from a Web of document to a Web of interlinked knowledge. While the state of the art LOD contains billion of triples describing millions of entities, it has only a limited number of schema information and is lack of schema-level axioms. To close the gap between the lightweight LOD and the expressive ontologies, we contribute to the complementary part of the LOD, that is, Linking Open Schema (LOS). In this paper, we introduce Zhishi.schema, the first effort to publish Chinese linked open schema. We collect navigational categories as well as dynamic tags from more than 50 various most popular social Web sites in China. We then propose a two-stage method to capture equivalence, subsumption and relate relationships between the collected categories and tags, which results in an integrated concept taxonomy and a large semantic network. Experimental results show the high quality of Zhishi.schema. Compared with category systems of DBpedia, Yago, BabelNet, and Freebase, Zhishi.schema has wide coverage of categories and contains the largest number of subsumptions between categories. When substituting Zhishi.schema for the original category system of Zhishi.me, we not only filter out incorrect category subsumptions but also add more finer-grained categories.

Keywords: Linking Open Data, Linking Open Schema, Integrated Category Taxonomy, Large Semantic Network

1 Introduction

With the development of Semantic Web, a growing amount of structured (RDF) data has been published on the Web. Linked Data [3] initiates the effort to connect distributed data across the Web. Linking Open Data (LOD)[3] is the largest community for semantic data publishing and interlinking. It converts the Web from a Web of document to a Web of knowledge. There have been over 200 datasets within the LOD project. Among these datasets, DBpedia [4], Yago [9], and Freebase [5] serve as hubs to connect others. More recently, Zhishi.me [11] has been developed as the first effort of Chinese LOD. It extracted RDF triples

[3] http://linkeddata.org/

from three largest Chinese encyclopedia Web sites, namely Baidu Baike, Hudong Baike, and Chinese Wikipedia. It also creates `owl:sameAs` links between two resources from different sources if these resources refer to the same entity.

While LOD contains billions of triples describing millions of entities, the number of schemas in LOD is limited. Yago defines explicit schema to describe concept subsumptions as well as domains and ranges of properties. Freebase has a very shallow taxonomy with domains and types. If we consider the schemas having labels in Chinese, the number is even smaller. Moreover, the qualities of schemas within these datasets are not always satisfactory. The DBpedia community creates the DBpedia Ontology project[4] which lets users define mapping rules to generate high-quality schema from ill-defined raw RDF data.

On the other hand, there exist some works to publish schema-level knowledge. Schema.org[5] provides a shared collection of schemas that webmasters can use to markup HTML pages in ways recognized by major search providers. However, it is manually created and does not have a Chinese version. BabelNet [10] is a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms in 50 languages. It is also a semantic network which connects concepts and named entities, made up of more than 9 million entries. Probase [12] is a universal probabilistic taxonomy which contains 2.7 million concepts harnessed automatically from a corpus of 1.68 billion Web pages. While it is the largest taxonomy, the usage of Probase is restricted in Microsoft. Meanwhile, the development of social media provides us a chance to create schema-level knowledge from folksonomies. A recent survey paper [8] compares different approaches of discovering semantics of tags. The main focus of these approaches is to capture the hierarchical semantic structure of folksonomies.

In this paper, we contribute to Linking Open Schema (LOS). LOS aims at adding more expressive ontological axioms between concepts. Links in LOS are created between concepts from different sources and are not limited to equivalence relations. More precisely, we introduce Zhishi.schema, the first effort to publish Chinese linked open schema. We collect navigational categories as well as dynamic tags from more than 50 most popular social Web sites in China. We then propose a two-stage method to capture equivalence, subsumption and relate relationships between the collected categories and tags. Compared with approaches to build a taxonomy from the tag space, Zhishi.schema additionally extracts `equal` and `relate` relations to form a large semantic network. Different from Probase, we publish Zhishi.schema as open data for public access. BabelNet is the closest work to ours. But it collects data from a small number of sources including WordNet, Open Multilingual WordNet, Wikipedia, OmegaWiki, Wiktionary, and Wikidata while Zhishi.schema extracts semantic relations between categories from a large number of popular Chinese social Web sites.

The rest of the paper is organized as follows. Section 2 gives an overview of our approach. Section 3 describes the technical details. Section 4 shows the experimental results of Zhishi.schema in terms of data size, quality, and coverage.

[4] `http://wiki.dbpedia.org/Ontology`
[5] `https://schema.org/`

Section 5 introduces Web access to Zhishi.schema and finally we conclude the paper in Section 6.

2 Overview

In this section, we start with a brief introduction of the problem, then list several challenges, and finally provide the overall process.

2.1 Problem Definition

Input: Given a set of Chinese social media Web sites $WS = \{ws_1, ws_2, \ldots, ws_n\}$, for each Web site ws, it might contain a set of categories $C_{ws} = \{c_1, c_2, \ldots, c_n\}$ as well as a set of tags $T_{ws} = \{t_1, t_2, \ldots, t_m\}$. These categories are organized in a hierarchical way. In a *category hierarchy*, a category might be associated with zero or several parent categories as well as child categories. We call c_i a *static category* as it is relatively stable and predefined by the Web site. The tags are organized in a flat manner. We call t_j a *dynamic category* because it is created on the fly by Web users. In fact, a tag can be treated as a single node category with no parents or children.

Output: We aim at building a Chinese linked open schema called *Zhishi.schema* composed of categories from the input Web sites. Zhishi.schema contains three types of semantic relations, namely `relate`, `subclassOf`, and `equal`. More precisely, two categories (no matter static or dynamic) are related if their meanings are close. One category is a subclass of another if and only if the former is a child of the latter. Two categories are equal if and only if they refer to the same meaning. The `relate` relation is the weakest semantic relation among the three types. All these semantic relations are asymmetric just like `owl:sameAs` in LOD. That is to say, c_1 `sr` c_2 is not identical to c_2 `sr` c_1 where c_1, c_2 are two categories, and `sr` $\in$ {`relate`, `subclassOf`, `equal`}. The `subclassOf` relations form an integrated concept taxonomy while the other two kinds of semantic relations build a large semantic network.

2.2 Challenges

As categories come from various sources, extracting semantic relations between categories is not a trivial task. In particular, we have the following challenges.

- *Incorrect hierarchy of static categories.* A category and its parents from the hierarchy of a Web site might dissatisfy the `subclassOf` relation. For instance, "Athlete" is defined as a parent category of "Athlete Type". Clearly, it indicates an incorrect subsumption relation. Therefore, the quality cannot be ensured if we directly treat the existing hierarchy of static categories as a part of the local site schema without any refinement.

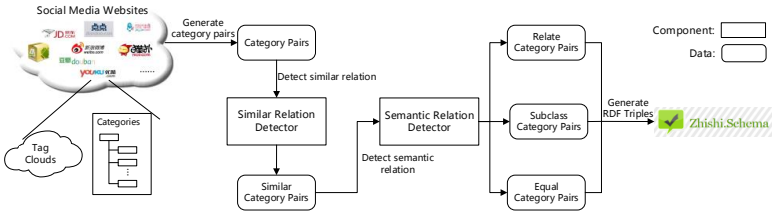


Fig. 1. The Workflow of Our Approach

– *Ambiguity of categories in different contexts.* If the label of a category refers to more than one meaning, the category becomes ambiguous. In another word, we cannot distinguish two categories sharing the common label if contexts are not taken into account. For example, "Apple" can be a kind of fruit or the Apple company. We cannot determine its exact meaning until it has a parent category labeled by "IT company". So it is quite important to consider context information when revealing the meaning of a category.
– *Lack of representation for categories* Unlike documents, categories do not have plenty of textual information to describe them. When detecting semantic relations between categories, current text mining techniques cannot be directly applied until we find a way to enrich the representation of categories.

2.3 Workflow

We now provide a workflow to explain the whole process and its components. As shown in Figure 1, we have two main components, namely *Similar Relation Detector* (*SimRD*) and *Semantic Relation Detector* (*SemRD*).

The input of Similar Relation Detector is category pairs generated from different Web sites. *SimRD* tries to filter out dissimilar pairs and feeds similar category pairs as the input of Semantic Relation Detector. Then *SemRD* identifies the semantic relation type (i.e. `relate`, `subclassOf`, or `equal`) of each similar category pair. These semantic relations are converted into RDF triples for building Zhishi.schema. Our approach is a two-stage method. In the first stage, we design "cheap" features to represent each category and use lightweight learning algorithms to find out similar pairs. This leads to a significant reduction of the number of category pairs and a much cleaner input for the second stage. We then represent categories with more sophisticated features and treat semantic relation detection as a multi-class classification problem to solve. The details of *SimRD* and *SemRD* will be introduced in the next section.

3 Approach

3.1 Similar Relation Detection

Category Representation The simplest way to represent a category c is using its category label $l(c)$. However, it is insufficient if the labels of two categories do

not have any overlapped words or share very few words. For example, "NYC" and "New York City" are synonyms, but their labels are totally different.

Inspired by Explicit Semantic Analysis (ESA) [7], we map a category into several concepts in a knowledge base, and then use these concepts to represent the category. The benefits are three-folds. First, the category representation is enriched from its label into a set of concepts. Second, the dimension of concepts is usually much lower than that of text features so that we avoid curse of dimensionality and enable efficient processing. Third, the concepts are higher-quality than texts with less ambiguities.

Here, Baidu Zhidao[6], the largest Chinese community QA site, is chosen to serve as the knowledge base. When submitting $l(c)$ as a keyword to Baidu Zhidao, we collect first 10 pages containing relevant questions. From these questions, their associated categories are obtained. These categories form the related concept set of c, denoted as $\mathrm{RCS}(c) = \{rc_1, rc_2, \ldots, rc_n\}$ where rc_i is the i-th related concept. We can further use them to define the related concept vector $\mathrm{RCV}(c)$ in form of $< rc_1(c), rc_2(c), \ldots, rc_n(c) >$ where $rc_i(c)$ stands for the occurrence of the related concept rc_i. The occurrence is the number of questions belonging to rc_i. It reflects the importance or popularity of rc_i. These two representations can help discover similar category pairs if two categories share a large portion of related concepts but vary a lot on their labels.

The key to the success of ESA lies on the coverage of the knowledge base and the quality of concept mapping. We tried every category from a collection of Web sites (see Section 4.1 for details), only 1.2% categories do not have any related concepts. Then we use Baidu Zhidao's own categories to test the mapping quality. For 14 root categories, 10 are the most occurred related concepts of themselves, and 4 are ranked second. For all categories (2118 in all), more than half are ranked in top three. Only 17% (366 categories) do not contain themselves in their related concept vectors. The above two tests show Baidu Baike has a wide coverage to return related concepts for most categories, and therefore suitable to be a knowledge base for concept mapping.

The label representation $l(c)$ and two variants of the related concept representation (i.e. $\mathrm{RCS}(c)$ and $\mathrm{RCV}(c)$) are called *local representations* of c. Besides, static categories are organized in a hierarchical way. Thus, a category c might have ancestors and descendants which can be treated as *neighbors* of the category. If we aggregate related concepts of these neighbors to $\mathrm{RCS}(c)$ and $\mathrm{RCV}(c)$, we get *enriched representations* of c, denoted as $\mathrm{RCS}^+(c)$ and $\mathrm{RCV}^+(c)$ respectively. $\mathrm{RCS}^+(c)$ only adds related concepts of c's neighbors which are not related concepts of c. $\mathrm{RCV}^+(c)$ not only counts the occurrence of the newly added related concepts, but also increases the occurrences of existing ones if they are related concepts of some neighbor of c. Compared with the local representations, the enriched ones further capture context information to represent the category, and thus can help disambiguate its meaning.

[6] `http://zhidao.baidu.com/`

Category Similarity Measures We apply some widely-used similarity measures to the above category representations.

- *Similarity based on category label.* This measure is actually string matching based on longest common substring (LCS)[7]. The similarity between categories c_1 and c_2 is defined as:

$$\text{CLsim}(c_1, c_2) = \frac{\text{LCS}(l(c_1), l(c_2))}{|l(c_1)|} \tag{1}$$

  Where $|l(c)|$ is the string length of c's label, and $\text{LCS}(l(c_1), l(c_2))$ is the longest common substring between $l(c_1)$ and $l(c_2)$.
- *Similarity based on related concept set.* This measure is actually the Jaccard similarity[8] between two sets. The similarity is defined as follows.

$$\text{RCSsim}(c_1, c_2) = \frac{|\text{RCS}(c_1) \cap \text{RCS}(c_2)|}{|\text{RCS}(c_1)|} \tag{2}$$

- *Similarity based on related concept vector.* This measure is based on cosine similarity[9] between two vectors, which is defined as:

$$\text{RCVsim}(c_1, c_2) = \frac{\sum_{rc \in \text{RCS}(c_1) \cap \text{RCS}(c_2)} rc(c_1) \cdot rc(c_2)}{\sum_{rc \in \text{RCS}(c_1)} rc(c_1)^2} \tag{3}$$

While the label-based string measure captures the linguistic similarity between two categories, the related concept based measures capture structural similarities between these categories. Thus, we consider all these three similarity measures to estimate the relatedness of a category pair. We treat these similarity measures as features and apply a machine learning algorithm to predicting whether the two categories are similar or not.

$$\text{P}_{sim}(c_1, c_2) = m(\text{CLsim}(c_1, c_2), \text{RCSsim}(c_1, c_2), \text{RCVsim}(c_1, c_2)) \tag{4}$$

Where m stands for some learning model and $\text{P}_{sim}(c_1, c_2)$ is the prediction probability. If $\text{P}_{sim}(c_1, c_2)$ is greater than a threshold, the two categories are considered to be similar. Analogously, we define $\text{RCSsim}^+(c_1, c_2)$ and $\text{RCVsim}^+(c_1, c_2)$ when the enriched representations are used. $\text{P}^+_{sim}(c_1, c_2)$ is further defined when $\text{RCSsim}^+(c_1, c_2)$, $\text{RCVsim}^+(c_1, c_2)$, and $\text{CLsim}(c_1, c_2)$ are used as features.

In fact, similar relation detection is a binary classification problem. We choose three most popular classification models for m. They are J48 Decision Tree, Logistic Regression, and Multi-Layer Perceptron. For all the three models, we use the implementations in Weka[10] with default parameter values to perform

[7] http://en.wikipedia.org/wiki/Longest_common_substring_problem
[8] http://en.wikipedia.org/wiki/Jaccard_similarity
[9] http://en.wikipedia.org/wiki/Cosine_similarity
[10] http://sourceforge.net/projects/weka/

experiments. For $P^{+}_{sim}(c_1, c_2)$, we need to decide which neighbors should be used for the enriched representations of c_1 and c_2. Here, we only consider the parents and children of a category in some hierarchy as its neighbors. This is because high-level ancestors and low-level descendants cannot represent the context of a category in a discriminative way. Also, the average depth of a category hierarchy is usually of a small value (See Table 1 in Section 4 for details). Moreover, some improper categories are placed as the parents or children of a category in some hierarchy for the purpose of Web site navigation only. To reduce the noise, we filter out neighbors if the probabilities of being similar with the category are low. More details and more experimental results will be discussed in Section 4.2.

3.2 Semantic Relation Detection

Textual Context based Category Representation Semantic relations are finer-grained similar relations. The above mentioned category representations are insufficient especially for tags to detect semantic relations. Thus, we leverage contextual words co-occurred with a category c frequently to represent the category. We call it the *textual context representation* of c, denoted as $\mathrm{TC}(c)$.

A category c might be associated with several pages in a Web site. We could use the contents of these pages for $\mathrm{TC}(c)$. However, the numbers of pages associated with different categories vary a lot. Moreover, pages from different sites differ in terms of content length and wording styles. For example, a tweet is much shorter than a news page and contains more informal language expressions.

Instead, we use text snippets returned by a search engine to represent a category. More precisely, we submit $l(c)$ as a keyword to the largest Chinese search engine Baidu[11] and return a list of relevant Web pages in form of snippets. Each snippet contains the page title, a small fraction of the page content with surrounding words of $l(c)$, and the link to the page. The snippets of top 20 search results are selected for further processing. After word segmentation and stopword removal, a set of terms are obtained to represent c as a "virtual" document. In our implementation, we use Ansj[12] as the Chinese word segmenter with a widely used stopword list in Chinese. We further adopt TF-IDF (short for Term Frequency-Inverse Term Frequency) [1] for term weighting. As a result, $\mathrm{TC}(c)$ is a n-dimension vector $< w_1(c), w_2(c), \ldots, w_n(c) >$ where the weight of the i-th term $\mathrm{TC}(c)_i$ is $w_i(c)$ and n is the number of all terms of all categories. If a term w does not co-occur with $l(c)$, the corresponding value in $\mathrm{TC}(c)$ is zero.

Category Similarity Measures We additionally define $\mathrm{TCsim}(c_1, c_2)$ to measure the *similarity based on textual context*:

$$\mathrm{TCsim}(c_1, c_2) = \frac{\sum_{i=1}^{n} \mathrm{TC}(c_1)_i \cdot \mathrm{TC}(c_2)_i}{\sum_{i=1}^{n} \mathrm{TC}(c_1)_i^2} \tag{5}$$

[11] `http://www.baidu.com`

[12] `https://github.com/ansjsun/ansj_seg`

We add this similarity measure as a new feature to a learning model for predicting the probability a certain kind of semantic relation holds. Since the prediction accuracy of $\mathrm{P}^{+}_{sim}(c_1, c_2)$ is higher that that of $\mathrm{P}_{sim}(c_1, c_2)$ for detecting similar relations no matter which learning model is used, we combine $\mathrm{TCsim}(c_1, c_2)$ with $\mathrm{CLsim}(c_1, c_2)$, $\mathrm{RCSsim}^{+}(c_1, c_2)$, and $\mathrm{RCVsim}^{+}(c_1, c_2)$ as follows.

$$\begin{aligned}\mathrm{P}_{sem}(c_1, c_2) = m(&\mathrm{CLsim}(c_1, c_2), \mathrm{RCSsim}^{+}(c_1, c_2),\\ &\mathrm{RCVsim}^{+}(c_1, c_2), \mathrm{TCsim}(c_1, c_2))\end{aligned} \tag{6}$$

Semantic relation detection is treated as a three-class classification problem where class labels are "relate", "subclassOf", and "equal". We use Support Vector Machine (SVM) for m with the Radial Basis Function (known as RBF) kernel implemented in Weka. In addition to the learning-based approach, we also propose a heuristic-based method as a baseline to detect semantic relations. For a similar category pair (c_1, c_2), if $l(c_1)$ is the same as $l(c_2)$, we create an `equal` relation. If $l(c_1)$ is the suffix of $l(c_2)$, a `subclassOf` relation is generated to indicate c_2 is a child category of c_1. After applying these two heuristic rules, the remaining similar category pairs are considered to have `relate` relations.

4 Experiments

4.1 Data Statistics

We select 51 popular social media Web sites in China. The data was crawled in December, 2013. The detailed statistics of each site are shown in Table 1. From the table, we list the site name, its URL, the site type, the category number, the tag number, and the average depth of the category taxonomy. If some site does not contain any category or tag, we use ⁄ to indicate the value of that column is missing. Since the semantics of tags are less stable than those of static categories. We do not take all tags from these sites to build Zhishi.schema. Instead, we only selected popular tags during last December. In total, we collected 408,069 labels in which 328,288 are categories and 79,781 are tags.

4.2 Accuracy Evaluation

We first carry out experiments on small labeled datasets to determine the optimal combination of category representations and the learning algorithms. The trained model having the best performance is then used to detect semantic relations on the whole dataset. Finally, an evaluation theme is introduced along with quality assessment results on Zhishi.schema.

Training on Small Labeled Datasets Classification is supervised learning, which requires labeled data for training. The classification performance depends on whether the labeled data is adequate and whether training data and test data have the similar distributions. In order to ease the burden of manual labeling and

Table 1. Statistics for 51 Popular Social Media Web Sites in China

Site	URL	Type	#Category	#Tag	Avg Depth
360 Mobile Phone Assistant	http://sj.360.cn/	App Market	49	/	1.69
91 Mobile Phone Assistant	http://zs.91.com/	App Market	76	/	1.55
Amazon	http://www.amazon.cn/	E-commerce	3,311	/	3.65
Android Market	http://apk.hiapk.com/	App Market	279	/	2.56
Apple App Store	http://www.apple.com/cn/	App Markets	90	/	1.69
Baidu Baike	http://baike.baidu.com/	Wiki	10,445	/	2.67
Baidu Tieba	http://tieba.baidu.com/	BBS	214	/	1.57
Baidu Wenku	http://wenku.baidu.com/	Document Sharing	299	/	1.87
Baidu Zhidao	http://zhidao.baidu.com/	Q&A	2,118	/	3.24
BaiXing	http://www.baixing.com/	Classified	55,179	/	4.08
DangDang	http://www.dangdang.com/	E-commerce	6,847	/	2.59
DianDian	http://www.diandian.com/	Light Blog	/	14,294	/
DingDing Map	http://www.ddmap.com/	Customer Review	34,142	/	2.64
Docin	http://www.docin.com/	Document Sharing	734	/	1.60
Douban	http://www.douban.com/	Social Network	13,172	/	4.04
FanTong	http://www.fantong.com/	Customer Review	3,842	/	2.61
XianGuo	http://xianguo.com/	RSS	38	/	1.62
GanJi	http://www.ganji.com/	Classified	25,274	/	3.81
Guang	http://guang.com/	Social E-commerce	299	/	2.61
Hudong Baike	http://www.baike.com/	Wiki	23,995	/	5.49
JiangNanQingYuan	http://www.88999.com/	Dating	153	/	2.02
ShiJiJiaYuan	http://www.jiayuan.com/	Dating	82	/	1.83
360buy	http://www.jd.com/	E-commerce	31,140	/	3.59
KaiXing	http://www.kaixin001.com/	Social Network	125	/	2.45
Lvping	http://www.lvping.com/	Online Travel	4,0475	/	3.57
MeiLiShuo	http://www.meilishuo.com/	Social E-commerce	316	/	2.57
Mop	http://www.mop.com/	BBS	25	/	1.57
PPS	http://www.pps.tv/	Video Sharing	814	/	1.67
QieKe	http://www.qieke.com/	LBS	6,224	/	3.51
QiongYou	http://www.qyer.com/	Online Travel	107	7,400	1.68
RenHe	http://www.renhe.cn/	Business Social Network	250	/	2.55
RenRen	http://www.renren.com/	Social Network	119	/	1.98
RenRen Game	http://wan.renren.com/	Social Gaming	43	/	1.70
RenRen XiaoZhan	http://zhan.renren.com/	Light Blog	/	7,038	/
RuoLin	http://www.wealink.com/	Business Social Network	62	/	1.56
Sina iAsk	http://iask.sina.com.cn/	Q&A	5,247	/	3.24
Sina Blog	http://blog.sina.com.cn/	Blog	27	16,190	1.56
Sina Game	http://games.sina.com.cn/	Social Gaming	54	/	1.67
Sina GongXiang	http://ishare.sina.com.cn/	Document Sharing	234	/	1.57
Sina Micro Blog	http://weibo.com/	Microblogging	184	/	2.66
TaoBao	http://www.taobao.com/	E-commerce	1,845	/	3.34
Tencent Blog	http://blog.qq.com/	Blog	24	/	1.65
Tencent Micro Blog	http://t.qq.com/	Microblogging	16	/	1.00
TianYa	http://www.tianya.cn/	BBS	1,769	/	3.18
Tudou	http://www.tudou.com/	Video Sharing	755	/	1.64
TuiTa	http://www.tuita.com/	Light Blog	/	5,122	/
Netease Blog	http://blog.163.com/	Blog	20	/	1.60
Netease Micro Blog	http://t.163.com/	Microblogging	/	29,737	/
Netease Reader	http://yuedu.163.com/	RSS	46	/	1.83
Chinese Wikipedia	http://zh.wikipedia.org/	Wiki	56,985	/	3.71
Youku	http://www.youku.com/	Video Sharing	744	/	1.62

to avoid distribution bias, we propose an effective method to create labeled data. To detect similar category pairs, the training data has two labels: "similar" as positive and "dissimilar" as negative. A category pair (c_1, c_2) is considered as a positive candidate if the arithmetic mean of $\text{CLsim}(c_1, c_2)$, $\text{RCSsim}(c_1, c_2)$, and $\text{RCVsim}(c_1, c_2)$ is above 0.5. Otherwise, the category pair is possibly negative. We randomly select positive and negative candidates in a uniform way from all the collected Web sites for further user verification. To build a labeled dataset for semantic relation detection, we evenly sample similar category pairs from all these sites and apply the heuristic-based method to generate possible labels. These labels are manually verified and revised accordingly.

We apply 5-fold cross validation to train models in all experiments. Note that K-fold cross validation is widely used in statistics to overcome the over-fitting problem. *Precision*, *recall*, and *F-measure* are used for effectiveness study. Precision is the fraction of retrieved category pairs that are relevant while recall

Table 2. Effectiveness Comparison between Local and Enriched Representations

Method	Precision		Recall		F-Measure	
J48 Decision Tree	0.777	**0.80**	0.754	**0.882**	0.765	**0.839**
Logistic Regression	0.767	**0.778**	0.736	**0.864**	0.751	**0.819**
Multi-Layer Perceptron	0.749	**0.783**	0.781	**0.922**	0.765	**0.847**

is the fraction of relevant category pairs that are retrieved. For similar relation detection, similar category pairs are relevant. For semantic relation detection, a category pair having a certain type of semantic relation is relevant. The F-measure (also known as F_1 score) is the harmonic mean of precision and recall.

- *Evaluating similar relation detection.* The dataset contains 1,986 category pairs in which 398 pairs are labeled as "similar" and 1,588 pairs are labeled as "dissimilar". We list the precision, recall, and F-Measure of different learning models using local representations trained on the labeled dataset on the left side in Table 2. From the table, we can see that the Multi-Layer Perceptron model performs best. In the case of enriched representation, we remove neighbors of a category if the prediction probabilities of being similar with the category are below 0.1. The prediction probability is given by the best model using local representations (i.e., Multi-Layer Perceptron). After filtering, 76.14% static categories have one or more parents while only 10.18% have children. The right side of Table 2 shows the evaluation results of using enriched representations. All three learning models achieve significant improvements when enriched category representations are used. Still, Multi-Layer Perceptron has the best accuracy performance. Thus, this model is used to find similar category pairs in all Web sites.
- *Evaluating semantic relation detection.* The training data has 800 similar category pairs. Among them, 500 are labeled as "relate", 240 are labeled as "subclassOf", and 60 are labeled as "equal". We compare three approaches (i.e. heuristic-based, learning-based, and their combination) in our effectiveness study. The combined approach first accepts `equal` relations and `subclassOf` relations found by the heuristic rules. For the remaining similar category pairs, it uses the learning-based approach for classification. Table 3 shows the evaluation results of three approaches for all kinds of semantic relations. From the table, we can see that the heuristic-based method performs better than the learning-based one when dealing with `equal` and `subclassOf` relations. This is because the heuristic-based one uses "hard" rules, which achieves very high precisions. The learning-based approach gets more promising results for `relate` relation detection since the heuristic-based one simply treats all remaining category pairs as `relate`, which brings more false positive examples. The combined one outperforms both approaches.

Accuracy of Three Semantic Relations in Zhishi.schema Zhishi.schema contains 1,560,725 `subclassOf` relations, 22,672 `equal` relations and 229,167 `relate` relations. Since there are no ground truths available, we have to verify

Table 3. Heuristic-based vs. Learning-based Approach

Relation	Method	Precision	Recall	F-Measure
relate	Heuristic-based	0.794	0.981	0.787
	Learning-based	0.861	0.938	0.898
	Combination	**0.894**	**0.947**	**0.914**
subclassOf	Heuristic-based	0.927	0.543	0.685
	Learning-based	0.695	0.489	0.574
	Combination	**0.854**	**0.606**	**0.709**
equal	Heuristic-based	0.958	0.857	0.905
	Learning-based	0.909	0.657	0.763
	Combination	**0.912**	**0.939**	**0.925**

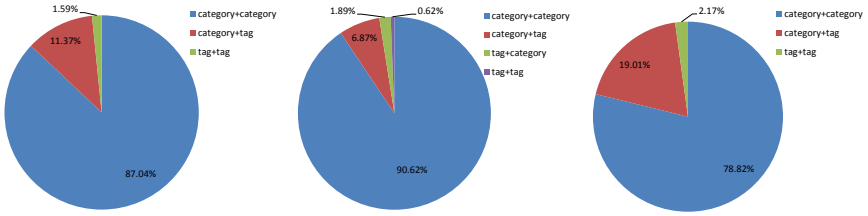
these relations manually. Due to the large number of semantic relations, it is impossible to evaluate all of them by hand. Therefore, we design an evaluation theme including a *sampling* strategy and a *labeling* process. Sampling aims to extract a subset of relations (called *samples*) which can represent the distribution of the whole result set. Then we can perform manual labeling to evaluate the correctness of samples. The accuracy assessment on samples are used to approximate the correctness of Zhishi.schema.

Sampling. For a kind of semantic relation `sr`, we study the relation distribution w.r.t. Web sites. A relation is of the form c_1 `sr` c_2 where c_1 and c_2 are categories. If c_1 or c_2 comes from a Web site, the Web site is treated as a *source* of the relation. A relation can have at most two sources. After iterating all relations of the same type, we can get the number of sources along with the relations in each source. For each source, we randomly select k relations. If k is greater than the total number of relations in the source, we take all of them for evaluation.

Labeling. We use the similar labeling process as that used in Yago. Four students participant in the labeling process. We provide them three choices namely *agree*, *disagree* and *unknown* to label each sample. After they label all the samples, we can compute the average accuracy. Finally, the Wilson interval [6] at $\alpha = 5\%$ is used to generalize our findings on the subset to the whole Zhishi.schema.

When applying the above evaluation theme, we get encouraging results.

- 50 Web sites contains `equal` relations. We randomly select 10 relations from each site and 487 samples are returned. After labeling, the average number of agree votes is 440, and the precision achieves $90.03\% \pm 2.63\%$.
- 45 sources have `relate` relations. We get 450 samples with $k = 10$. The average number of agree votes is 404, and the precision is $89.44\% \pm 2.80\%$.
- Compared with the flat structure of `equal` or `relate` relations, `subclassOf` relations form a hierarchical acyclic graph (HAG). The root depth is 1 and the maximal depth is 16. Since a category may have one or more parents, we can traverse to the category from the roots via different paths. These paths might have different lengths so that each category could exist at multiple depths of HAG. On average, the depth of each category is 3.479. In order for comprehensive evaluation, we need to cover every source at each depth of HAG. When sampling at a depth ranging from 2 to 16, k is set to 5. As a result, we get 2,922 `subclassOf` relations for manual labeling. The average number of agree votes is 2,456, and the final precision is $84.01\% \pm 1.33\%$.



(a) relate (b) subclassOf (c) equal

Fig. 2. Category Pair Pattern Distribution in Three Types of Semantic Relations

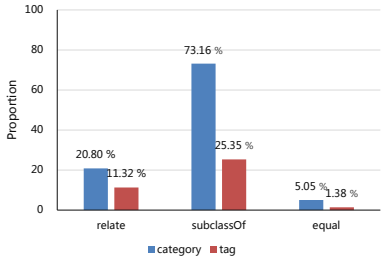


Fig. 3. Relation Proportions

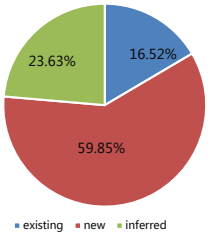


Fig. 4. Subclass Distribution

4.3 Data Distribution of Zhishi.schema

Category pairs can be grouped into three patterns, namely `category+category`, `category+tag`, and `tag+tag`. For `subclassOf` relation, we divide `category+tag` into two sub-patterns. `tag+category` indicates a tag is a sub concept of a category while `category+tag` means a category is a sub concept of a tag. Figure 2 shows the category pair pattern distribution in all three types of semantic relations. From the figure, we can see `category+category` contributes to the largest proportion (more than 75%) of relations for any kind of semantic relation. In contract, less than 5 percent come from `tag+tag`. The huge gap can be explained by the semantic stability of categories and the ambiguity nature of tags.

As shown in Figure 3, 73.16% categories (25.35% tags) appear in `subclassOf` relations, 20.80% (11.32% respectively) contribute to `relate` relations, and 5.05% (1.38% respectively) for `equal` relations. The high proportion of `subclassOf` relations among categories (tags) enables Zhishi.schema to form a large concept taxonomy. The ratio of `equal` relations is pretty low because it is the most strict semantic relation and thus similar category pairs seldom satisfy such relation.

We also check the number of `subclassOf` relations already defined in some category hierarchies. As shown in Figure 4, the proportion of existing subsumptions is 16.52%. Another 23.63% `subclassOf` relations can be inferred from category hierarchies via intermediate paths. Notice that 59.85% new `subclassOf` relations are discovered, which shows the value of Zhishi.schema.

Table 4. Overlap between Zhishi.schema and Other Datasets

	Zhishi.schema	DBpedia	Yago	BabelNet	Freebase
Category Number	408,069	142,139	49,407	619,226	2,035
Overlap with Zhishi.schema	/	82,586	24,036	23,193	567
Subclass Number	**1,560,725**	3	256,538	55,486	1,092
Subclass Overlap with Zhishi.schema	/	2	34,354	2,762	79

4.4 Comparison With Other Datasets

Overlap of Categories and Subsumptions We compare Zhishi.schema with other well-known datasets namely DBpedia[13], Yago[14], BabelNet[15] and Freebase[16] in terms of categories and subclasses. Table 4 shows the category and subclass information of each dataset. It also lists the category overlap and subclass overlap between Zhishi.schema and the other datasets. As for the category number, Zhishi.schema is larger than DBpedia, Yago and Freebase. It also contains half of the categories from DBpedia and Yago. In BabelNet, a category corresponds to a synset. Since many synsets contain Chinese labels, BabelNet has the largest number of categories. Regarding `subclassOf` relations, Zhishi.schema has the largest number (six times larger than the second largest one – Yago). You may find that there are only 3 `subclassOf` relations in DBpedia. Ontological subsumptions are only defined in the DBpedia ontology while the ontology does not contain a Chinese version. So we leverage the multilingual nature of Wikipedia and finally get three `subclassOf` relations with both sides having the Chinese correspondences. When looking at the subclass overlap, we find only small overlaps between Zhishi.schema and the other datasets. Thus, combining Zhishi.schema with these datasets could form a larger linked open schema.

Overlap of Equivalence Relations with BabelNet Zhishi.schema contains 22,672 `equal` relations where 4,380 of them represent the same meaning with different labels. BabelNet is the largest multilingual semantic network in the world. For each conept in BabelNet, it is organized in form of a synset in which there are synonyms representing the same concept in different labels or languages. Therefore, we would like to check how many extracted `equal` relations are covered by BabelNet. Here, we do not count a `equal` relation when categories in a pair have the same string. In this way, we get 1,270 `equal` relations covered by BabelNet. Due to the small overlaps of both `subclassOf` and `equal` relations, Zhishi.schema and BabelNet can complement with each other.

Refining Zhishi.me Category System Since Zhishi.schema includes all three Chinese encyclopedia sites (used for Zhishi.me), the resulting concept taxonomy comprises categories and category subsumptions in these three sites. Hence,

[13] `http://dbpedia.org/About`
[14] `http://www.mpi-inf.mpg.de/yago-naga/yago/`
[15] `http://babelnet.org/`
[16] `https://www.freebase.com/`

we can compare Zhishi.schema with Zhishi.me[17] to see how many incorrect `subclassOf` relations are filtered out and how many new `subclassOf` relations are discovered. We have developed two variants of Zhishi.schema: basic refined Zhishi.me category system (*Basic*) and enriched Zhishi.me category system (*Enriched*). Basic is obtained by collecting categories in Zhishi.me and `subclassOf` relations between these categories from Zhishi.schema. It only considers `subclassOf` relations in form of c_1 `subclassOf` c_2 where c_1 and c_2 belong to categories in Zhishi.me and c_1 is the direct child of c_2. Enriched further considers `subclassOf` relation paths with one or more intermediate categories from other sites in Zhishi.schema. The original Zhisih.me category system contains 251,160 `subclassOf` relations. Basic removes 211,386 `subclassOf` relations and adds 29,177 ones. Enriched additionally increases 69,776 `subclassOf` relations.

5 Web Access to Zhishi.schema

Besides the application of Zhishi.schema to refine the existing category system of Zhishi.me, we also provide online Web access for Zhishi.schema. Moreover, we allow users to download the data dump to build their own applications.

5.1 Linked Data

According to the Linked Data principles[18], Zhishi.schema creates URIs for all categories and provides sufficient information when someone looks up a URI by the HTTP protocol. Since Zhishi.schema contains categories from different sites, we design a URI pattern to indicate where a category comes from and whether it is static or dynamic. The pattern `http://zhishi.schema/[site]/[category type]/[label]` comprises of fours parts. `http://zhishi.schema/` is the namespace. The second part tells the provenance of the category. If it is a tag, the third part is `dynamic`. Otherwise, it is `static`. The last part is the category label.

When publishing Zhishi.schema, we follow the best practice recipes [2] and try to reuse existing RDF vocabularies which have clear semantics and are widely used. Particularly, we use `skos:related` for `relate` relations, `rdfs:subClassOf` for `subclassOf` relations, and `owl:equivalentClass` for `equal` relations. When Semantic Web agents that accept "application/rdf+xml" content type access our server, resource descriptions in the RDF format will be returned.

5.2 Lookup Service

We provide a lookup service for users to access Zhishi.schema. The service is available at `http://los.linkingopenschema.info/LookUp.jsp`. Given a query, all categories whose labels exactly match the query are returned. If two categories are `equal`, they are automatically merged as an integrated view for browsing.

[17] `http://zhishi.me/`
[18] `http://www.w3.org/DesignIssues/LinkedData.html`

Fig. 5. An Example Page of Integrated Categories

If a user searches for "Water Purifier", as shown in Figure 5, we return a page integrating two equivalent categories from two e-commerce Web sites (i.e. 360buy and DangDang). From the page, we can see provenances of two categories, other equivalent categories with different labels, their parent categories, child categories, related categories, and links to their original pages in Web sites. These information are organized in the `Resource Site Label`, `EqualClass`, `SuperClass`, `SubClass`, `RelatedClass` and `Link` sections respectively.

We can click on any parent category or child category to switch to another page view. Such an interaction stands for navigation in the integrated concept taxonomy of Zhishi.schema. A click on one related category or an equivalent category corresponds to traversal on the semantic network of Zhishi.schema.

5.3 SPARQL Endpoint

We also provide a SPARQL endpoint for querying Zhishi.Schema. Professional users can submit customized queries at `http://los.linkingopenschema.info/SPARQL.jsp`. We use AllegroGraph RDFStore[19] as the backend triple store.

6 Conclusions and Future Work

In this paper, we introduced Zhishi.schema, the first effort of publishing Chinese linked open schema. It contains an integrated concept taxonomy. It also comprises a large semantic network composed of `equal` relations and `relate` relations. Thus, Zhishi.schema can be a good start point to serve as the Chinese version of schema.org. Moreover, since Zhishi.schema reuses RDF and OWL vocabularies, it can be imported into any ontology editor for further refinement.

[19] `http://www.franz.com/agraph/allegrograph/`

As for future work, we will apply our approach to social media Web sites in other languages especially in English. The resulting dataset can be further linked with Zhishi.schema to form a multilingual linked open schema. We also plan to publish links between categories in Zhishi.schema and other data sources in LOD so as to build a global LOS.

Acknowledgements

This work was partially funded by the National Science Foundation of China through project No: 61272378. We thank Jianhao Li, Shaowei Ling, Wen Rui, and Yishu Fang for helping label the data. We also thank Chengyuan Xue for his valuable feedback and detailed comments during the proofreading process.

References

1. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM press New York (1999)
2. Berrueta, D., Phipps, J., Miles, A., Baker, T., Swick, R.: Best practice recipes for publishing rdf vocabularies. Working draft, W3C (2008)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. International journal on semantic web and information systems 5(3), 1–22 (2009)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)
5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. ACM (2008)
6. Brown, L.D., Cai, T.T., DasGupta, A.: Interval estimation for a binomial proportion. Statistical Science pp. 101–117 (2001)
7. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI. vol. 7, pp. 1606–1611 (2007)
8. Garcia-Silva, A., Corcho, O., Alani, H., Gomez-Perez, A.: Review of the state of the art: Discovering and associating semantics to tags in folksonomies. The Knowledge Engineering Review 27(01), 57–85 (2012)
9. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence 194, 28–61 (2013)
10. Navigli, R., Ponzetto, S.P.: Babelnet: Building a very large multilingual semantic network. In: Proceedings of the 48th annual meeting of the association for computational linguistics. pp. 216–225. Association for Computational Linguistics (2010)
11. Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., Yu, Y.: Zhishi.me - weaving chinese linking open data. In: International Semantic Web Conference (2). pp. 205–220 (2011)
12. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: A probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. pp. 481–492. ACM (2012)