Domain Ontology Learning Enhanced by Optimized Relation Instance in DBpedia

Liumingjing Xiao[1], Chong Ruan[1], An Yang[2], Junhao Zhang[1], Junfeng Hu[1, *]
[1] Key Laboratory of Computational Linguistics, Ministry of Education, Peking University, Beijing, P. R. China.
[2] School of Life Sciences, Peking University, Beijing, P. R. China.
{xlmj, pkurc, yangan, junhao.zhang, hujf}@pku.edu.cn

Abstract

Ontologies are powerful to support semantic based applications and intelligent systems. While ontology learning are challenging due to its bottleneck in handcrafting structured knowledge sources and training data. To address this difficulty, many researchers turn to ontology enrichment and population using external knowledge sources such as DBpedia. In this paper, we propose a method using DBpedia in a different manner. We utilize relation instances in DBpedia to supervise the ontology learning procedure from unstructured text, rather than populate the ontology structure as a post-processing step. We construct three language resources in areas of computer science: enriched Wikipedia concept tree, domain ontology, and gold standard from NSFC taxonomy. Experiment shows that the result of ontology learning from corpus of computer science can be improved via the relation instances extracted from DBpedia in the same field. Furthermore, making distinction between the relation instances and applying a proper weighting scheme in the learning procedure lead to even better result.

Keywords: Ontology Learning; Text Mining; DBpedia

1. Introduction

Ontologies are one of the most important parts of semantic webs and attract more and more attention. Domain ontology describes knowledge of a certain domain and has been used in many applications such as QA systems. However, constructing ontologies by hand is a tedious and time-consuming work.

In recent years, many ontology learning algorithms have been proposed to address this problem. These algorithms are designed to mine ontologies from unstructured, semi-structured or structured texts using statistical or linguistic methods. Clustering algorithms, which are based on the assumption that similar words have similar contexts, are typically used in statistical approach. Linden and Piitulainen (2004) used clustering to find synonyms and achieved good performance. Term subsumption is another useful technique, which employs conditional probabilities of the occurrence of terms to decide what extent the generality of a term is and thus discovers term hierarchy. Fotzo and Gallinari's (2004) work is an exemplar of this approach. In the linguistics settings, syntactic structure analysis and dependency analysis are utilized to discover hierarchical relations between terms. The idea of using lexico-syntactic patterns was proposed by Hearst (1998), who used some predefined patterns such as "NP such as {NP, NP … , (and | or)} NP" to obtain hypernyms. Many researchers followed this idea. To name just a few: Hippisley et al. (2005) used head-modifier principle to discover hypernym-hyponym pairs, and Sombatsrisomboon et al. (2003) exploited frames such as "X is a/an (kind of) Y" to finish the same task.

Besides these learning-from-free-text methods, some researchers start from a small, core ontology and enrich it with external data sources. Gavankar et al. (2012) presented work done towards populating a domain ontology using a public knowledge base (KB) like DBpedia. Booshehri's (2015) work is also a good example of this method.

In this paper, we propose an ontology learning method following He's (2014) method, which followed the well-known Ontology Learning From Text (OLFT) cake layer framework (Cimiano, 2006). We implement DBpedia as knowledge base to facilitate the ontology learning procedure. Different from all the algorithms mentioned above, which learn ontologies only from free texts or aim to populate or enrich ontologies using various data sources. We use DBpedia as a supervisory tool to guide the clustering process, so that the structure of learned ontology is more similar to Wikipedia concept hierarchy. This is the key contribution of our work: learning the structure itself during training procedure rather than populating the structure as a post-processing step.

Our method comprises two phases. In the first phase, we learn different types of term relations to enrich DBpedia datasets. Then, in the second one, the datasets are utilized to supervise ontology learning procedure: adjusting term distributional similarities by their relation instances in DBpedia.

The rest of this paper is organized as follows. Section 2 describes our ontology learning method in detail. The description of new language resources presented by this work can be found in section 3. Experimental results and analysis are in section 4. Finally, in section 5, we give conclusions and look forward to further researches.

2. Ontology Learning Method

According to the OLFT cake layer framework (Cimiano, 2006), the ontology learning task is actually divided into eight increasingly complex subtasks: terms, synonyms, concepts, concept hierarchy, relations, relation hierarchy, axiom schemata, and general axioms. Our ontology learning method focuses on the first four layers from the

* To whom all correspondence should be addressed.

bottom.

2.1 DBpedia Datasets

We use two of DBpedia datasets[1] from April 2015 as our knowledge base, namely, Wikipedia taxonomy and Wikipedia article tags. Wikipedia taxonomy is comprised of categories, which form a hierarchical structure and contain few cycles. We limit Wikipedia taxonomy within areas of computer science. Cycles in the original taxonomy are removed by deleting bottom-up edges, yielding a Directed Acyclic Graph (DAG) structure. Wikipedia articles are also limited by their category tags. We combine the two datasets by inserting articles into Wikipedia taxonomy as leaf nodes. For each article, edges from its tagged categories to this article are added into edge set of taxonomy. This generates a 19-layer Wikipedia concept hierarchy in computer science areas, including 4953 categories and 99398 articles.

To extract reliable hyponymy relations from Wikipedia concept hierarchy, we perform a greedy algorithm that obtains a Wikipedia concept tree structure from DAG structure. This algorithm sequentially assigns each node under its deepest parent node in the tree in topological order. In contrast with DAG structure, tree structure contains more reliable parent-child pairs, which are more likely to form hypernym-hyponym relations.

We also tag totally 38 types of attributes to nodes in Wikipedia concept tree, using top-down paths from high level category nodes in Wikipedia concept hierarchy. Attributes we choose are researchers, conferences, organizations, or other types of categories for concept classification, such as "theoretical computer scientists" and "20th-century software".

2.2 Relation Knowledge Base

We build a term relation knowledge base from term bank in computer science areas. Specifically, we cluster different kind of relations between terms into groups. A term relation is either suffix pattern (e.g., "computer" and "computer theory") or prefix pattern (e.g., "science" and "computer science"). From this relation knowledge base, we manually classify those suffix patterns with high frequency into two groups. One group contains common suffix patterns that prefer to generate true hypernym-hyponym relations. The other group contains suffix patterns that prefer to generate "pseudo" hyponymy relations that could not be referred to as common sense hyponymy relations. For example, "knowledge engineering" is constructed by appending suffix "engineering" behind "knowledge", while it is not a hyponym of "knowledge". Table 1 shows the classification between true hyponymy suffixes and "pseudo" hyponymy suffixes.

The term bank we build for relation knowledge base learning is in areas of computer science and contains 102421 terms in both English and Chinese. It is extracted from proposals of National Natural Science Foundation of China (NSFC) between 2009 and 2015. We establish a

True hyponymy suffixes	array attack characteristic classification communication compensation detection distribution extraction fusion learning localization mapping matching monitoring navigation prediction recognition reconstruction scheduling segmentation sensing sensor signal robot tracking transmission image imaging
"Pseudo" hyponymy suffixes	algorithm analysis assessment construction control data design device feature framework function effect environment estimation engineering evaluation generation information interface coding measurement mechanism method model modeling management network optimization pattern platform problem process processing project protocol scheme simulation software strategy structure system technique technology test testing theory security computing

Table 1: True hyponymy suffixes and "pseudo" hyponymy suffixes classification.

Chinese-English term mapping using bilingual keywords given by applicants in each proposal.

To narrow the gap between domain corpus and Wikipedia concepts, we add automatic generated hypernym-hyponym pairs from bilingual term bank to enrich Wikipedia concept tree contents. Term pairs with prefix or suffix pattern relations are inserted into Wikipedia concept tree as parent-child pairs. Those term pairs whose semantic similarities are lower than a certain threshold are ignored. Finally, we insert 3077 term pairs.

Another optimization to Wikipedia concept tree is discriminating true hyponymy from "pseudo" hyponymy using our relation knowledge base, which will be discussed later in section 2.3.

2.3 Domain Ontology Learning

This paper follows He's (2014) ontology learning method. He adopts a distributional similarity based method to discover semantically similar terms. Each term is represented by a distributional feature vector. Each dimension of the vector is the Point-wise mutual information (PMI) of corresponding feature. Then, the distributional similarity between two terms *ti* and *tj* is defined as cosine similarity between their corresponding distributional feature vectors PMI_{ti} and PMI_{tj}, i.e.:

$$Sim_{ti,tj} = \cos\langle PMI_{ti}, PMI_{tj} \rangle$$

Then we implement optimized relations in DBpedia to enhance ontology learning method. According to relation knowledge base mentioned above, we assort relations in Wikipedia concept tree into three groups, namely, "pseudo" hyponymy relations, true hyponymy relations, and brother relations. "Pseudo" hyponymy relations are direct parent-child pairs in Wikipedia concept tree that are

[1] http://wiki.dbpedia.org/Downloads2015-04

recognized as "pseudo" hyponymy by relation knowledge base, no matter if these pairs are inserted in the enrichment step or exist in the original Wikipedia taxonomy or article tags. True hyponymy relations are other direct parent-child pairs without "pseudo" relation attribute mark, and they are helpful for supervising hyponymy concept relations mining. Brother relations are concept pairs in Wikipedia concept tree that sharing the same direct parent node. These three types of relations are used as prior knowledge instances to supervise domain ontology learning process in this paper. Specifically, we increase each distributional similarity by different weight according to its original value and implemented relation instance type between terms. Formally, the modified term similarity is defined as follows:

$$Sim^{R}_{ti,tj} = \cos\left(r_{ti,tj}\langle PMI_{ti}, PMI_{tj}\rangle\right)$$

Here $r_{ti,tj}$ is the relation type weight between *ti* and *tj*. We choose to multiply relation weight by distributional feature vector angle instead of adding weight to distributional similarity directly. This prevents term pairs with high distributional similarity from growing too large to affect other terms similarities unusually, and is also helpful for data smoothing. The lower the term pair similarity is, the more strongly it is enhanced by the prior knowledge.

Then we follow He's (2014) method that apply a Hypertext-Induced Topic Search (HITS) and K-means based method to learn hierarchical domain ontology from unstructured text.

3. Description of Our Language Resources

We use the proposal set of NSFC between 2009 and 2013 as raw corpus to learn ontologies, and use DBpedia datasets as external knowledge base. Both are limited within areas of computer science.

Language resources	Number of layers	Number of nodes
Wikipedia concept tree	19	107428
Ontology	15	103125
Gold standard	5	5529

Table 2: Number of layers and nodes in Wikipedia concept tree, ontology, and gold standard.

We have constructed three language resources that are publicly available[2]. Firstly, we enrich the Wikipedia concept tree using bilingual term bank and relation knowledge base. All "pseudo" hyponymy relations in Wikipedia concept tree, including term pairs we generate from bilingual term bank, are marked with "other" relation attribute. Secondly, we learn computer science domain ontology enhanced by optimized relation instances in Wikipedia concept tree using method above. Finally, we also build a gold standard from NSFC taxonomy to

[2] http://icl.pku.edu.cn/Ontology_and_Wiki_of_ComputerScience.rar

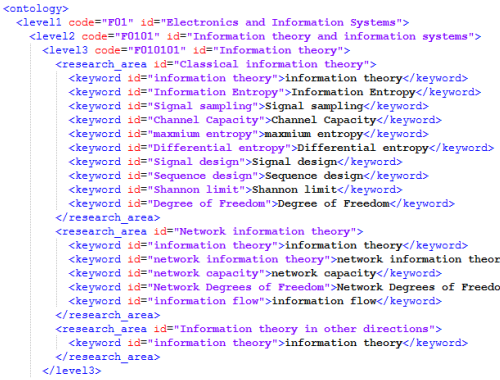

Figure 1: A snapshot of the gold standard.

evaluate automatic learned ontologies, which will be introduced in the next section. The number of layers and nodes of these language resources are listed in Table 2.

4. Evaluation

To illustrate the effectiveness of our method, we carry out experiments to evaluate learned ontologies. Due to the lack of a mature ontology evaluation platform, we design a metric to compare the learned ontology with a gold standard, which is from NSFC research area taxonomy and constructed by experts. The structure of the gold standard is demonstrated in Figure 1.

The gold standard comprises 5 layers: the first four layers are research areas, from the most general ones to the most specific ones. The bottom layer contains keywords in level-4 research area.

Due to the huge gap between the size of the learned ontologies and gold standard (see Table 2), we only evaluate the "recall rate" of near-synonyms in the learned ontologies. To be precise, we take all pairs of near-synonyms (two keywords under the same fourth layer research area) from the gold standard, and calculate their weighted distance in the learned ontology. Then, the average weighted distance is used as a measure of how close these near-synonyms are in the learned ontology. Definitely, the smaller the average distance is, the better the learned ontology is.

The distance between two concepts is calculated as follows:

Definition 1 (Entropy Weighted Edge Length): For an edge $e = (u, v)$ in a tree T, where node u is the parent of node v, the entropy weighted edge length of e is defined as log(number of children of u).

This definition is quite straight forward: if some node u has many children, it suggests node u represents a general concept and thus its children are not that similar to each other. In fact, the logarithm weighting scheme is also used in previous research (Rusu et al., 2014).

Definition 2 (Revised Entropy Weighted Distance): For two near-synonym *n1* and *n2* in the gold standard G, we denote the set of all their siblings (including themselves) by S. Given an ontology O, the revised entropy weighted distance between *n1* and *n2* is defined as:

$$d(n1, n2) = \max\{0, \sum_{e\ in\ path(n1,n2)} w(e) - 2\log_2|S|\}$$

Where $w(e)$ is the entropy weighted edge length of edge e,

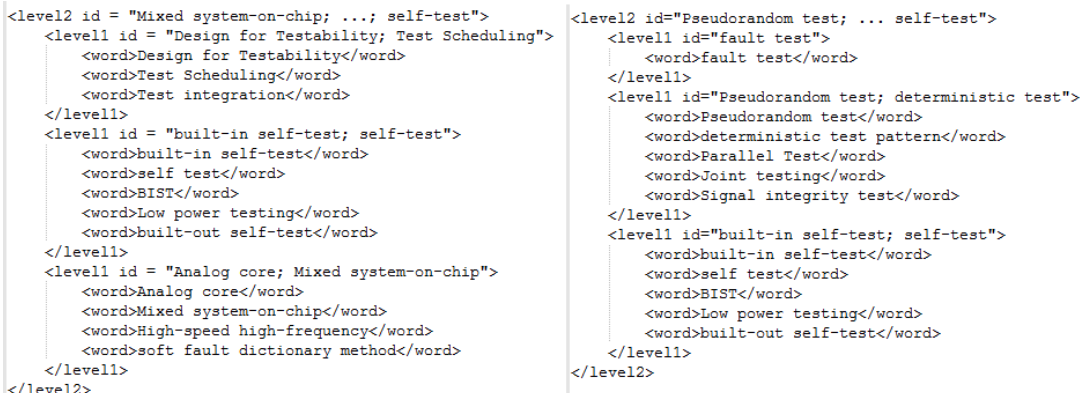

Figure 2: External knowledge helps improve ontology structure.

and $|S|$ denotes the cardinality of set S.

With the definition above, we have $d(n1,n2)$ always non-negative. What's more, $-2log|S|$ serves as a normalize factor, so that calculating the distance between $n1$ and $n2$ in gold standard G itself will always get zero. This paper use the revised entropy weighted distance to evaluate learned ontologies.

We design two baselines to compare with the ontology learned using our method. The first baseline is He's ontology learning method. In the second baseline, we implement DBpedia relations to supervise He's ontology learning process, but do not make distinction between relation types. After experiments, the optimal parameters for relation weights of true hyponymy, "pseudo" hyponymy, and brother relations in our method are 0.6, 0.9, and 0.8, respectively, which are 0.8, 0.8, and 0.8 in the second baseline. Experimental results are listed in Table 3.

Ontology learning method	Number of layers	Revised entropy weighted distance
He	14	22.06
He + DBpedia	14	21.46
He + DBpedia + relation KB	14	**21.29**

Table 3: Revised entropy weighted distance of three ontology learning methods.

Compared with He's method, implementing DBpedia relations obtains a decrement of 0.6 on the revised entropy weighted distance, and which is 0.77 for the proposed method. The results show that external concept knowledge base helps improve performance of domain ontology learning from unstructured text, and discriminating different types of relation instances contributes to a more reliable domain ontology structure. From our method, true hyponymy instance increases more weight on distributional similarities, while "pseudo" hyponymy instance affects the weight much less. This helps cluster hyponymy terms in lower layers during ontology learning.

In order to gain some intuition of the proposed method, we attach a small piece of the learned ontologies by He's and our method in Figure 2.

The ontology in the left is the baseline generated by He's method, while the right one is produced by our method. It is clear that in the baseline ontology there exists some terms that are not relevant enough, such as "high-speed high-frequency". And this "noise" disappears in the right ontology, which is due to its knowledge acquired from DBpedia relation instances. In fact, in the gold standard, the corresponding research area consists of 5 terms: pseudorandom test, built-in self-test, built-out self-test, On-chip test clock, and on-chip measurement, which is more similar to the ontology generated by the proposed method and is consistent with our intuition.

5. Conclusion

This paper proposes an ontology learning method that using DBpedia as a supervisory tool to guide the clustering process. Three language resources in areas of computer science are constructed in this work: enriched Wikipedia concept tree, domain ontology, and gold standard from NSFC taxonomy. Experiment shows that the result of ontology learning from corpus of computer science can be improved via the relation instances extracted from DBpedia. Furthermore, making distinction between the relation instances and applying a proper weighting scheme in the learning procedure lead to even better result. These mean that external knowledge base helps improve performance of domain ontology learning, and the ontology structure learned from our method is more reliable and similar to knowledge base we implement.

Further researches may include the following aspects. Firstly, more reliable hyponymy relations between concepts can be mined, in both quantity and quality. Secondly, the ontology learning method presented in this paper only focuses on the first four layers of OLFT cake layer framework. We intend to learn non-taxonomy relations between concepts in future work.

6. Acknowledgements

This research is supported by the National Natural Science Foundation of China (grant No. M1321005, 61472017).

7. Main References

Booshehri, M., Luksch, P. (2015). An Ontology Enrichment Approach by Using DBpedia. In *Proceedings of*

the 5th International Conference on Web Intelligence, Mining and Semantics. ACM.

Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Secaucus, NJ: Springer-Verlag New York, Inc.

Fotzo, H.N., Gallinari, P. (2004). Learning generalization specialization relations between concepts—application for automatically building thematic document hierarchies. In *Proceedings of the 7th International Conference on Computer-Assisted Information Retrieval (RIAO)*.

Gavankar, C., Kulkarni, A., Li, Y. and Ramakrishnan, G. (2012). Enriching an Academic Knowledge base using Linked Open Data. In *Proceedings of the Workshop on Speech and Language Processing Tools in Education*, COLING, pp. 51--60.

He, S., Zou, X., Xiao, L. and Hu, J. (2014). Construction of Diachronic Ontologies from People's Daily of Fifty Years. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, pp. 3258--3263.

Hearst, M. (1998). Automated discovery of WordNet relations. In *WordNet: An Electronic Lexical Database and Some of its Applications*, C. Fellbaum (Eds.). MIT Press, Cambridge, MA, pp. 131--152.

Hippisley, A. R., Cheng, D. and Ahmad, K. (2005). The head-modifier principle and multilingual term extraction. *Natural Language Engineering*, 11(2), pp. 129--157. Cambridge University Press.

Linden, K., Piitulainen, J. (2004). Discovering synonyms and other related words. In *Proceedings of the 3rd International Workshop on Computational Terminology*, CompuTerm, pp. 63--70.

Rusu, D., Fortuna, B. and Mladenić, D. (2014). Measuring concept similarity in ontologies using weighted concept paths. *Applied Ontology*, 9(1), pp. 65--95.

Sombatsrisomboon, R., Matsuo, Y. and Ishizuka, M. (2003). Acquisition of hypernyms and hyponyms from the WWW. In *Proceedings of the 2nd International Workshop on Active Mining*. Japan.