An Embedding-based Approach to Constructing OWL ontologies

Lijing Zhang[1,3], Xiaowang Zhang[1,3,*], Leyuan Zhao[1,3], Jiachen Tian[1,3], Shizhan Chen[1,3], Hong Wu[2,3], Kewen Wang[2,3], and Zhiyong Feng[2,3]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
[2] School of Computer Software, Tianjin University, Tianjin, China
[3] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
[*] Corresponding author.

Abstract. This paper presents a novel system OWLearner for automatically extracting axioms for OWL ontologies from RDF data using embedding models. In this system, ontology construction is transformed to the classification problem in machine learning and thus off-the-shelf tools can be employed to learn axioms in OWL. There are mainly three modules, namely, *embedding*, *sampling*, and *training & learning*. Large ontologies DBpedia and YAGO are used to validate the proposed approach. The experimental results show that OWLearner is able to learn high-quality expressive OWL axioms automatically and efficiently.

1 Introduction

An ontology is a formal representation of objects and their relationships in a domain of interest. OWL, with its latest version OWL 2, is the W3C standard for ontology languages. Formally, an ontology is expressed as a pair of an RDF dataset and a TBox.

Automatic construction of ontologies is an important but challenging task in ontology engineering. Specifically, given an RDF data, the task of constructing ontologies we are interested in is to extract DL axioms. DL-Learner [1] is a leading system for enriching DL ontologies, which is based on techniques, such as refinement operator, in inductive logic programming (ILP) [2]. However, ILP-based systems are usually unable to handle very large ontologies.

We tackle this challenge by providing a scalable method for learning DL axioms using machine learning techniques. Based on the embedding methods in representation learning, an RDF dataset is embedded into a continuous vector space and the inherent structure of the original data is preserved [3]. Thus, the ontology construction can be accomplished in the vector space via supervised machine learning, which in essence is to learn a function for each axiom pattern to predict the correctness of input axioms. To achieve this goal, labeled samples are obtained through SPARQL queries, and are transformed into vector space using embedding and feature engineering techniques.

In this paper, we have implemented a system prototype OWLearner and compared it with the state of art system DL-Learner on major benchmarks such as DBpedia and YAGO for the DL axiom constructing task. Our experimental results show that OWLearner outperformed DL-Learner in both time efficiency and the quality of axioms.

2 An Overview of OWLearner

In this section, we introduce OWLearner, a prototype system for learning axioms in description logics (DL). Specifically, we consider the 12 axiom patterns in $\mathcal{SROIQ}$, which are popular [4], listed as follows: $P_1 : C \sqsubseteq D$; $P_2 : C \equiv D$; $P_3 : s \sqsubseteq r$; $P_4 : s \equiv r$; $P_5 :\geq 1\, r \sqsubseteq C$; $P_6 : \top \sqsubseteq \forall r.C$; $P_7 : r_1 \circ r_2 \sqsubseteq s$; $P_8 : \exists r.C \sqsubseteq D$; $P_9 : C \equiv D \sqcap E$; $P_{10} : C \equiv D \sqcap \exists r.E$; $P_{11} : C \sqsubseteq \exists r.(\exists s.D)$; $P_{12} : s \equiv r^{-}$.

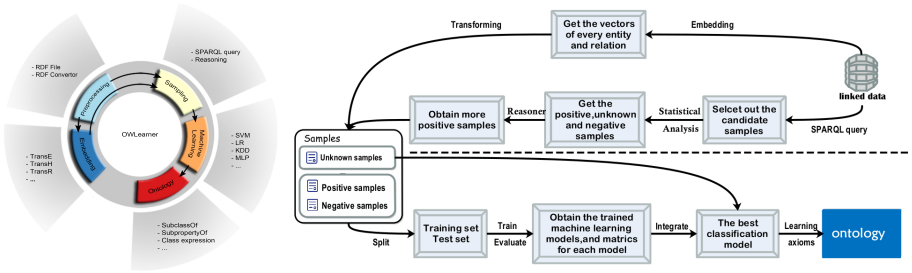

Fig. 1. Framework and workflow of OWLearner

The framework and workflow of OWLearner are shown in the left and right subfigure of Figure 1, respectively. The OWLearner contains five components as follows:

Data preprocessing This component specifies how to retrieve data and how to convert various formats of RDF data to a unified format (e.g., N-Triples) so that most RDF serialization formats are supported conveniently.

Embedding This component embeds entities and relations into continuous vector spaces as their features by employing effective embedding models, such as TransE [5]. The selection mechanism of embedding models depends on the scores of benchmark datasets. Then, embeddings of axioms are constructed by applying some feature engineering methods based on the original embeddings.

Sampling This component generates labeled samples for training, which fall into three parts: positive samples, negative samples, and unknown samples to be further judged. This procedure of generating samples consists of three steps: SPARQL querying, statistical analysis, and OWL reasoning (rule-based and ontology reasoning).

Learning This component trains supervised learning models via positive/negative samples and then predicts axioms in unknown samples by applying the trained models. OWLearner supports most of the supervised learning models, and provides many principal metrics such as accuracy (ACC) and Area Under ROC curve (AUC) to evaluate these models. Moreover, we define some metrics to evaluate the learned axioms, including *Standard Confidence*(*SC*), *Head Coverage*(*HC*) and *Partial Completeness Assumption(PCA) based Confidence*(*PCAconf*).

Building This part constructs all axioms generated/predicted in our model as an OWL ontology. OWLearner provides a plugin API which supports most off-the-shelf ontology editors.

3 Experiments and Evaluation

In this section, we evaluate OWLeaner on three data sets, namely, DBpedia, YAGO1k (a fragment of YAGO containing all classes with over 1000 entities), and Chinese Symptom Database (SIC) shown in Table 1. We conducted four sets of experiments and explain them in detail as follows.

Table 1. Specification of datasets

Data	DBpedia	YAGO1k	SIC
Entity	6099488	4295827	73812
Relation	659	38	19
Fact	18154761	12430700	617486
Class	14989	4987	16

Table 2. Precision of P_1 to P_6

Axiom	P_1	P_2	P_3	P_4	P_5	P_6
Precision	0.88	0.73	0.33	0.24	0.78	0.71

Table 3. Performance of classifiers for learning 12 axioms

Axiom	rate	unknown	learned	ACC	AUC	Classifier
P_1	0.99	3049	199	0.82	0.83	GBDT
P_2	0.99	5972	817	0.75	0.80	GBDT
P_3	0.9	5630	827	0.83	0.63	SVM
P_4	0.7	5658	524	0.86	0.88	SVM
P_5	0.9	14434	5138	0.82	0.90	GBDT
P_6	0.7	25346	4349	0.82	0.88	GBDT
P_7	0.9	45941	4294	0.81	0.77	GBDT
P_8	0.9	34081	10565	0.84	0.91	GBDT
P_9	0.99	59866	7576	0.88	0.92	DT
P_{10}	0.99	1036709	6817	0.95	0.92	DT
P_{11}	0.99	1032741	19237	0.95	0.96	KNN
P_{12}	0.99	3097	79	0.94	0.99	SVM

Set 1. Suitability of Classifiers based on ACC and AUC In this set of experiments, for each of those 12 axiom patterns, we tested which machine learning model is most effective based on two metrics. The results are shown in Table 3, which show that no single learning model is most effective for all axiom patterns. This experiment provides a guideline for selecting a suitable learning model for a given axiom pattern.

Set 2. Accuracy of learned axioms We used three metrics, namely, *Standard Confidence(SC)*, *Head Coverage(HC)* and *PCA-based Confidence(PCA-conf)*, to test the quality (accuracy) of learned axioms by OWLearner. The results are shown in Table 4, which indicates that OWLearner can learn axioms with high quality.

Set 3. Precision of learned axioms We used DBpedia as the benchmark to evaluate the precision of OWLearner. As only axioms of patterns $P_1, \ldots, P_6$ allows in DBpedia, the precisions for such axioms are obtained. The precision value for each axiom pattern represents the proportion of the axioms that can be matched. The results are shown in Table 2. The results show that relatively high precisions are obtained for axiom patterns P_1, P_2, P_5 and P_6, but the precisions of P_3 and P_4 are low. A major reason for this is that there is little data available for these two axiom patterns.

Set 4. Comparison of OWLearner with DL-Learner In this set of experiments, we compared the performance of OWLearner with DL-Learner based on four metrics *Runtime*, *Standard Confidence(SC)*, *Head Coverage(HC)* and *PCA-based Confidence(PCA-conf)*. The results, shown in Table 5, indicate that the quality of learned axioms by OWLearner are comparable to that by DL-Learner, although OWLearner is superior to DL-Learner in terms of *HC* degree. The major advantage of OWLearner is time efficiency. OWLearner does not need to specify a class name but DL-Learner requires to specify a target class before axioms can be learned.

Table 4. Accuracy of OWLearner in learning 12 axioms

Axiom	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
SC	0.97	0.86	0.61	0.70	0.70	0.63	0.84	0.88	0.88	0.72	0.63	0.38
HC	0.33	0.77	0.4	0.72	0.47	0.32	0.42	0.31	0.87	0.67	0.5	0.25
PCAconf	0.98	0.88	0.78	0.87	0.89	0.81	0.90	0.9	0.88	0.86	0.83	0.58

4 Conclusions

We have proposed a novel method of learning axioms for OWL/DL axioms. The method is based on the technique of embedding in representation learning. Based on the proposed method, we have implemented a system OWLearner for automatic axiom extraction in OWL ontologies. Our experiments show that OWLearner is much more efficient than DL-Learner, state of the art system for ontology axiom learning, and the quality of learned axioms for these two methods is comparable.

Table 5. Comparision between OWLearner and DL-Learner

Class	OWLearner				DL-Learner			
	Runtime	*SC*	*HC*	*PCAconf*	Runtime	*SC*	*HC*	*PCAconf*
$Library$	15min (total)	1.0	0.89	1.0	8min	1.0	0.51	1.0
$Guitarist$		1.0	0.87	1.0	9min	1.0	0.66	1.0
$Album$		1.0	0.8	1.0	20min	1.0	0.43	1.0
$SoccerPlayer$		1.0	0.78	1.0	10min	1.0	0.64	1.0
$RadioStation$		1.0	0.89	1.0	9min	1.0	0.73	1.0

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61502336), the National Key R&D Program of China (2016YFB1000603,2017YFC0908401), and the Seed Foundation of Tianjin University (2018XZC-0016).

References

1. L. Bühmann, J. Lehmann, and P. Westphal. (2016). DL-Learner: A framework for inductive learning on the Semantic Web. *J. Web Sem.*, 39:15–24.
2. S. Muggleton, L.De Raedt, D. Poole, I. Bratko, P. Flach. (2012) ILP turns 20. *J. Machine Learning.*, 86(1): 3–23.
3. M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich. (2016). A review of relational machine learning for knowledge graphs. *J. Proceedings of the IEEE*, 104(1): 11–33.
4. L. Bühmann, J. Lehmann. (2013) Pattern based knowledge base enrichment. *Proc. ISWC'13*, pp. 33–48.
5. A. Bordes, N. Usunier, J. Weston, O. Yakhnenko. (2013). Translating embeddings for modeling multi-relational data. *Proc. of NIPS'13*, pp. 2787–2795.