Enriching Ontologies through Data

Mahsa Chitsaz*

School of Information and Communication Technology,
Griffith University, Australia
mahsa.chitsaz@griffithuni.edu.au

Abstract. Along with the vast usage of ontologies in different areas, non-standard reasoning tasks have started to emerge such as concept learning which aims to drive new concept definitions from given instance data of an ontology. This paper proposes new scalable approaches in light-weight description logics which rely on an inductive logic technique in favor of an instance query answering system.
Keywords. OWL Ontology, Light-weight Description Logics, Concept Learning, Enriching Ontology.

1 Problem Description

Along with the vast use of DLs ontologies, non-standard reasoning tasks have started to emerge. One of such tasks is concept learning which is a process to find a new concept description from assertions of an ontology. The concept learning system plays an essential role in ontology enrichment as well as ontology construction. Ontology enrichment from unstructured or semi-structured data is an onerous task even for knowledge engineers. Additionally, the new added information may have diverse presentations among different engineers. As an example of concept learning, if a data set includes the assertions (John enrolled in the Semantic Web course) and (John is a Student), then a concept of "Student" can be learned by this data set which is "Who enrolled in at least one course". Therefore, this new concept definition inducted by the data will enrich the terminology of the ontology.

The current approaches of concept learning [11, 9, 20] are mostly presented for expressive DLs that are not scalable in practice. Since there are large practical ontologies that are represented by less expressive DLs such as the SNOMED CT[1], and the Gene ontology[2], it is plausible to propose a learning system for light-weight DLs that are tractable fragments of DLs in regards to standard reasoning tasks. The dedicated reasoners of light-weight DLs, such as CEL [1], Snorocket [17], and ELK [13] are very efficient for ontologies with only a TBox. These off-the-shelf reasoners do not fully support the ABox reasoning which is essential in the learning framework.

* Principal Supervisor: Professor Kewen Wang
[1] http://www.ihtsdo.org/snomed-ct/
[2] http://www.geneontology.org/

Therefore, the main research question is how to propose a learning framework to efficiently and scalably construct a concept description in light-weight description logics such as DL $\mathcal{EL}^+$ and DL-Lite. In fact, there are two main objectives for this research. The first is to design a scalable learning system which can work with real world ontologies. The second objective is to maximize the accuracy of a learned concept having incompleteness in data sets.

The remainder of this paper is organized as follows. Some preliminaries are presented in Section 2. In the next Section, the related work is investigated to find its limitations. In Section 4, the accomplished work to partially tackle the concept learning problem is presented and in Section 5, future plan followed by the evaluation of the proposed learning framework is discussed.

2 Preliminaries

An ontology in DLs consists of a *terminology* box, *TBox* $\mathcal{T}$, which represents the relationship among concepts and properties and an *assertion* box, *ABox* $\mathcal{A}$, which preserves the instances of the represented concepts and properties.

OWL EL[3], which is based on DL $\mathcal{EL}^+$ [2], is suitable for applications employing ontologies that contain very large numbers of properties and classes. In DL $\mathcal{EL}^+$, concept descriptions are inductively defined using the following constructors: $\top|\bot|\{a\}|C \sqcap D|\exists r.C$, where C and D are concept names, r is a role name, and a is an individual. An $\mathcal{EL}^+$-TBox includes general concept inclusions (GCIs) $C \sqsubseteq D$ and role inclusions (RIs) $r_1 \circ \ldots \circ r_k \sqsubseteq r$.

The DL-Lite family [5] is a family of light-weight description logics, which introduced for efficient query answering over ontologies with a large ABox, that is, the basis formalism of OWL QL[4]. Concepts and roles in DL-Lite$_R$ are constructed according to the following syntax: $B \rightarrow A|\exists R \qquad R \rightarrow P|P^-$ $C \rightarrow B|\neg C|C_1 \sqcap C_2 \qquad E \rightarrow R|\neg R$, where A denotes an atomic concept, P an atomic role, and P^- the inverse of atomic role P. B denotes a basic concept, that is either an atomic concept or a concept of the form $\exists R$. A DL-Lite$_R$ TBox is constructed by a finite set of inclusion assertions of the form $B \sqsubseteq C$ and $R \sqsubseteq E$, where B, C, R, and E are defined as above.

Note that normalized $\mathcal{EL}^+$-TBox only consists of these axioms: $A_1 \sqcap A_2 \sqsubseteq B$, $A \sqsubseteq \exists r.B$, $\exists r.A \sqsubseteq B$, $r_1 \circ \cdots \circ r_k \sqsubseteq r \in \mathcal{T}$, where $k \leq 2$, A, A_i and B are atomic concepts or $\top$. Then every existential quantifier $A \sqsubseteq \exists r.B$ in $\mathcal{EL}^+$-TBox can be replaced by these DL-Lite axioms $\{A \sqsubseteq \exists s, \exists s^- \sqsubseteq B, s \sqsubseteq r\}$.

3 Related Work

Concept learning in DLs concerns learning a general hypothesis from the given examples of a background knowledge that one wants to learn. Aiming to find a description of a *goal* concept G, there are two kinds of examples: positive

[3] `http://www.w3.org/TR/owl2-profiles/#OWL_2_EL`
[4] `http://www.w3.org/TR/owl2-profiles/#OWL_2_QL`

examples E_G^+, which are instances of G, and negative examples E_G^-, which are not. Literally, an example set of G, $\mathcal{A}'$, is a subset of ABox, $\mathcal{A}$; that is $\mathcal{A}' = \{G(a_1),\ G(a_2),\ \ldots,\ G(a_p),\ \neg G(b_1),\ \neg G(b_2),\ldots,\ \neg G(b_n)\}$, consequently $E_G^+ = \{a_1, a_2, \ldots, a_p\}$ and $E_G^- = \{b_1, b_2, \ldots, b_n\}$.

Example 1. By considering the following ABox, positive and negative examples:
$\mathcal{A} = \{\text{hasChild}(\textit{John, Chris}), \text{hasChild}(\textit{Mary, Chris}), \text{hasChild}(\textit{Joe, John}),$
$\text{Male}(\textit{John}), \text{Female}(\textit{Mary}), \text{Male}(\textit{Joe}), \text{Male}(\textit{Chris})\}$
$E_G^+ = \{\textit{Joe, John}\}$ $E_G^- = \{\textit{Mary, Chris}\}$.
A possible answer of the concept learning problem of the goal concept "Father" is $\exists \text{hasChild} \sqcap \text{Male}$.

Currently, most of the approaches to concept learning for DLs are an extension of inductive logic programming (ILP) methods. In the area of concept learning in description logics, promising research has been investigated and described in [11, 9, 20]. All of these approaches have been proposed for expressive DLs such as $\mathcal{ALC}$. One of the most significant concept learning system for DLs is DL-Learner [20] which has different heuristics to explore the search space with a built-in instance checker to employ *Close World Assumption* (CWA), that is faster than standard reasoners. However, none of these are scalable to work with real world ontologies. Nevertheless, there is little research on concept learning in DLs that transfer DL axioms to *logic program*s (LP), then apply the ILP method in order to learn a concept [10]. On the one hand, this approach is too expensive in terms of computation time. On the other hand, it is not always guaranteed that this conversion is possible. Additionally, another approach to tackle the concept learning problem in DLs is by employing a *Machine Learning* approach such as Genetic Programming [18] and kernels [8]. The experimental results of these approaches show that longer concept descriptions are generated compared with ILP based methods.

In terms of learning a concept description in less expressive DLs, research is limited. A learner for DL $\mathcal{EL}$, proposed by Lehmann and Haase [19], uses minimal trees to construct DL $\mathcal{EL}$ axioms then refines these by refinement operators. The DLs axioms were converted to trees and four different operators were defined to refine these trees. Apart from those ILP-based approaches, Rudolph [24] proposed a method based on Formal Concept Analysis (FCA) to generate a hypothesis. Further Baader et. al. [3] have used FCA to complete a knowledge base. Both of these methods used a less expressive DLs, where the former used $\mathcal{FLE}$, and the latter used a fragment of DLs which is less expressive than $\mathcal{FLE}$. These approaches demand many interactions of a knowledge engineer as an oracle of the system which is not applicable in most scenarios. In future plan, an automated system to learn new concept definitions more efficiently will be developed.

The above-mentioned approaches mostly focused on concept learning in expressive DLs, where it is not possible to have a scalable learner due to the fact that the underlying reasoners are not scalable. Therefore, a learner which produces a concept description in DL $\mathcal{EL}^+$ will be proposed, and can be employed

for DL-Lite ontologies. In the preliminary research, a learner system for DL $\mathcal{EL}^+$ using ILP-based approach and reinforcement learning technique was introduced.

4 Research Accomplished

In this section, an $\mathcal{EL}^+$ learner has been proposed since the current approaches aim to construct a concept definition in expressive DLs. However, an $\mathcal{EL}^+$ ontology necessitates the learned concepts expressed in $\mathcal{EL}^+$ only. This concept learning system is based on *inductive logic program* (ILP) techniques and finds a concept definition in $\mathcal{EL}^+$ through a *refinement operator* and *reinforcement learning* [6].

Concept Learning System using Refinement and Reinforcement: An effective tool to build the search space of concept hierarchies is requiered. According to the previous research in ILP, a refinement operator is suitable for this purpose. The proposed system benefits from the strength of the current refinement operators for $\mathcal{ALC}$ [10, 20], and a refinement operator for $\mathcal{EL}$ [19]. Downward (upward) refinement operators construct specializations (generalizations) of hypotheses [23]. The pair $\langle F, R\rangle$ is a *quasi-ordered set*, if a relation R on a set F is reflexive and transitive. If $\langle F, \sqsubseteq\rangle$ is a quasi-ordered set, a *downward refinement operator* for $\langle F, \sqsubseteq\rangle$ is a function ρ, such that $\rho(C) \subseteq \{D | D \sqsubseteq C\}$. For example, a subset of $\rho(\top)$ in the Example 1 is {Male, Female, $\exists$hasChild}, and a subset of $\rho(\exists$hasChild) is {Male $\sqcap$ $\exists$hasChild, Female $\sqcap$ $\exists$hasChild, $\exists$hasChild.Male, $\exists$hasChild.Female}. Since the refinement operator can build all possible mutations of concepts and roles, finding a correct concept description could not happen by a simple search algorithm, unless an external heuristic was employed to traverse the search space effectively. We have done some preliminary experiments in employing *reinforcement learning* (RL) technique in pruning the search space. In the proposed system, a state of a hypothesis is how correct this hypothesis is w.r.t. the given examples. This is found by the Pellet reasoner[5]. Initially, the hypothesis is the $\top$ concept. Then, an RL agent will change the hypothesis by choosing one action among those possible member of downward refinements of current hypothesis. The definition of actions is based on refinement operators that specializes the hypothesis to cover more positive examples and less negative examples. The correctness of the hypothesis, which is a score for the RL agent, will be determined by finding the instances of it. A signal is given to the RL agent according to its score to lead it to the goal state which the hypothesis is a solution of the concept learning problem. The possible actions for each state guide the RL agent to achieve the goal by this systematic reward-based approach. This approach shows promising results, however choosing an action is a non-deterministic task that causes problem where the given example sets are incomplete.

[5] `http://clarkparsia.com/pellet`

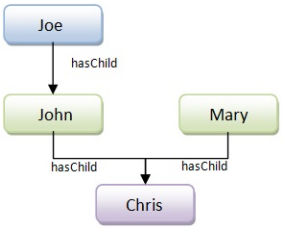
5 Future Plan

Most of the current approaches in the concept learning, including the proposed system in Section 4, use DL reasoners to accomplish instance checking task except DL-Learner which has a built-in instance checker. As a result of using OWL reasoners for the learning framework, the system becomes unscalable. Therefore, employing an efficient instance query answering (IQA) system is important for the learning framework. In this approach, query answering system is employed in order to compare certain answers of the constructed concept definition (as a query) with the given examples. A bottom-up algorithm [4] is efficiently constructed the hypothesis space, then the accuracy of any constructed concept is checked by the IQA system. An instance query (IQ) is of the form $C(x)$ with C either an $\mathcal{EL}^+$-concept or DL-Lite concept depends on learning a concept in $\mathcal{EL}^+$ or DL-Lite respectively.

Firstly, an IQA system will be developed for $\mathcal{EL}^+$ and DL-Lite queries. To achieve this, it is essential to understand how the current query answering system works efficiently. It is well-known that pure query rewriting [14] approaches are inefficient because of the exponential blow-up of the query size. Then query rewriting with auxiliary symbols [15] is introduced to include some auxiliary symbols to make the rewriting in polynomial time and this approach necessitates the saturated ABox. Our IQA is inspired by [22, 16], which complete the ABox into a canonical model $\mathcal{I}_\mathcal{K}$ of the ontology in polynomial time and independently from the input query. When $\mathcal{I}_\mathcal{K}$ can be constructed in polynomial time w.r.t. the size of the ontology, one can answer all instance queries of concepts or roles in the ontology signature efficiently. However, those auxiliary symbols cannot be the certain answer of any IQs, therefore, these unnamed individuals will be filtered from the result set.

Concept Learning System using Instance Query System: In this approach, the constructed canonical interpretation is employed as a fundamental tool to use a bottom-up algorithm in constructing a concept definition. The second research target is to construct consequence sets [12] of all positive and negative examples which are derived by IQA system. More precisely, a consequence set of an individual $a \in \text{ind}(\mathcal{A})$ is a pair $\langle rlist, clist\rangle$, where $rlist \subseteq N_R$ and $clist \subseteq N_C$ such that $\exists b \in \Delta^{\mathcal{I}_\mathcal{K}}$ with $\forall r \in rlist$, $(a,b) \in r^{\mathcal{I}_\mathcal{K}} \vee (b,a) \in r^{-\mathcal{I}_\mathcal{K}}$, and $\forall C \in clist$, $b \in C^{\mathcal{I}_\mathcal{K}}$. Then, all consequence sets of an individual a are combined as a consequence node, which is a pair $\langle rootset, conset\rangle$ such that $rootset = \{C | \mathcal{K} \models C(a)\}$ and $conset$ is the set of all consequence set of individual a. In Figure 1, the consequence nodes of the ABox instances in Example 1 are shown. Therefore, for all members of E_G^+ and E_G^-, the consequence hierarchy is constructed in order to induct a concept description. In our running example, the concept "Father" is constructed based on the common part of the consequence nodes for both *Joe* and *John* as positive examples, which in this case is "Male $\sqcap$ $\exists$hasChild", or "Male $\sqcap$ $\exists$hasChild.Male", although the second solution is subsumed by the first answer. As another example, if one wants to find a definition of the concept "Parent" with positive examples of *Joe*, *John* and *Mary*, and negative example of *Chris*, the common part of all those positive ex-

amples are ∃hasChild or ∃hasChild.Male which are correct concept descriptions for the given ontology and the example sets. Since the main interest is to find a shortest concept description, if in the first step of constructing consequence nodes a definition can not be learned, i.e. "Grandparent" in Example 1, this consequence node is extended to another step for positive examples until there is a unique common part for all consequence node of positive examples which does not overlap with any consequence node of negative examples.



Instances	Consequence Node
John	$\langle\{\text{Male}, \top\}, \{\langle\{\text{hasChild}\}, \{\text{Male}, \top\}\rangle, \langle\{\text{hasChild}^-\}, \{\text{Male}, \top\}\rangle\}\rangle$
Joe	$\langle\{\text{Male}, \top\}, \{\langle\{\text{hasChild}\}, \{\text{Male}, \top\}\rangle\}\rangle$
Mary	$\langle\{\text{Female}, \top\}, \{\langle\{\text{hasChild}\}, \{\text{Male}, \top\}\rangle\}\rangle$
Chris	$\langle\{\text{Male}, \top\}, \{\langle\{\text{hasChild}^-\}, \{\text{Female,Male}, \top\}\rangle\}\rangle$

Fig. 1. All first-step consequence nodes of the ABox instances of Example 1

6 Evaluation

The preliminary work on concept learning has been evaluated on family ontology from DL-Learner data sets[6] which is artificially constructed for test purpose and is smaller than practical ones. The proposed approach will be evaluated against current concept learning systems such as DL-Learner and YinYang[7]. There is no common benchmark for evaluating the ontology learning, although test cases have been borrowed from Machine Learning community[8] and transferred to DLs ontologies in data sets from [20]. All data sets from these concept learning systems will be used in the evaluation of the proposed approach. There are two main challenges in these benchmarks. First of all, most of the ontologies are expressed in expressive DLs, and solutions of a learning problem is not expressible by an $\mathcal{EL}$-concept description. Secondly, the second aim of this research is to have a scalable learning framework which these data sets are not applicable since the largest ontology has less than a million ABox assertions. Therefore, the LUMB benchmark[9] will be used to work on millions of ABox assertions. Some concept definitions will be removed from the TBox, then the proposed concept learning system will be applied to learn these missing concepts, and learned definitions are compared with their initial definitions. Therefore, the 'gold standard' for the

[6] http://sourceforge.net/projects/dl-learner/files/DL-Learner/
[7] http://www.di.uniba.it/∼iannone/yinyang/
[8] http://archive.ics.uci.edu/ml/
[9] http://swat.cse.lehigh.edu/projects/lubm/

learning problems are produced by querying the benchmark before the change. The completeness degree of the LUMB data sets will be tuned by another data generator [21].

As another evaluation plan, the proposed approach will be evaluated by the SNOMED CT ontology that contains more than 300K concept names, and around 60 role names in order to assess the scalability of the learning framework. However, the SNOMED CT ontology is only included a TBox which is the case for most of real world ontologies. Therefore, an ABox will be generated, for example by having different instances for all concept and role names. Then the proposed learning approach will be evaluated the same way as mentioned for the LUMB ontology by removing some definitions from the original ontology. There is also a general way of evaluating ontology learning [7], which those different metrics as quantitative evaluations will be employed in the evaluation plan.

7 Conclusion

In this paper, the concept learning problem is described to introduce its possible application in ontology enrichment. Then, two different approaches are presented for concept learning in light-weight description logics in Section 4 and Section 5. The preliminary results obtained on a small data set are encouraging which will lead to an improvement of the prototypical system to build a scalable learner. A fundamental tool to check the correctness of a learned concept definition is an instance checking system, subsequently an instance query answering system will be deployed in the proposed approach. Future work includes an implementation of the proposed approach in Section 5, as well as evaluating the scalability and efficiency of the proposed learning framework as mentioned in Section 6.

References

1. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL—A Polynomial-time Reasoner for Life Science Ontologies. In: Proceedings of the 3rd International Joint Conference on Automated Reasoning (2006)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence. pp. 364–369 (2005)
3. Baader, F., Ganter, B., Sattler, U., Sertkaya, B.: Completing Description Logic Knowledge Bases using Formal Concept Analysis. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. pp. 230–235 (2007)
4. Baader, F., Sertkaya, B., Turhan, A.Y.: Computing the Least Common Subsumer w.r.t. a Background Terminology. Journal of Applied Logic 5(3), 392 – 420 (2007)
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family. Journal of Automated Reasoning 39, 385–429 (2007)
6. Chitsaz, M., Wang, K., Blumenstein, M., Qi, G.: Concept Learning for $\mathcal{EL}^{++}$ by Refinement and Reinforcement. In: Proceedings of the 12th Pacific Rim International Conference on Artificial Intelligence (2012)

7. Dellschaft, K., Staab, S.: On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: Proceedings of the 5th international conference on The Semantic Web (2006)
8. Fanizzi, N., d'Amato, C.: A Declarative Kernel for $\mathcal{ALC}$ Concept Descriptions. In: The 16th International Symposium on Foundations of Intelligent Systems (2006)
9. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL Concept Learning in Description Logics. In: The 18th International Conference on Inductive Logic Programming (2008)
10. Fanizzi, N., Ferilli, S., Iannone, L., Palmisano, I., Semeraro, G.: Downward Refinement in the ALN Description Logic. In: The 4th International Conference on Hybrid Intelligent Systems. pp. 68–73 (2004)
11. Iannone, L., Palmisano, I., Fanizzi, N.: An Algorithm Based on Counterfactuals for Concept Learning in the Semantic Web. Applied Intelligence 26(2), 139–159 (2007)
12. Kaplunova, A., Möller, R., Wandelt, S., Wessel, M.: Towards scalable instance retrieval over ontologies. In: Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management (2010)
13. Kazakov, Y., Krötzsch, M., Simancík, F.: Concurrent Classification of EL Ontologies. In: Proceedings of the 10th International Conference on The Semantic Web (2011)
14. Kikot, S., Kontchakov, R., Zakharyaschev, M.: On (in) tractability of OBDA with OWL 2 QL. In: Proceedings of the 23th International Workshop on Description Logics (2011)
15. Kikot, S., Kontchakov, R., Podolskii, V.V., Zakharyaschev, M.: Long Rewritings, Short Rewritings. In: Proceedings of the 2012 International Workshop on Description Logics (2012)
16. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The Combined Approach to Query Answering in DL-Lite. In: Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (2010)
17. Lawley, M., Bousquet, C.: Fast Classification in Protégé: Snorocket as an OWL 2 EL Reasoner. In: Proceedings of the 6th Australasian Ontology Workshop (Advances in Ontologies) (2010)
18. Lehmann, J.: Hybrid Learning of Ontology Classes. In: Machine Learning and Data Mining in Pattern Recognition (2007)
19. Lehmann, J., Haase, C.: Ideal Downward Refinement in the $\mathcal{EL}$ Description Logic. In: The 20th International Conference on Inductive Logic Programming (2010)
20. Lehmann, J., Hitzler, P.: Concept Learning in Description Logics using Refinement Operators. Machine Learning 78(1-2), 203–250 (2010)
21. Lutz, C., Seylan, I., Toman, D., Wolter, F.: The Combined Approach to OBDA: Taming Role Hierarchies using Filters. In: Proceedings of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (2012)
22. Lutz, C., Toman, D., Wolter, F.: Conjunctive Query Answering in the Description Logic EL using a Relational Database System. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (2009)
23. Nienhuys-Cheng, S.H., Wolf, R.d.: Foundations of Inductive Logic Programming. Springer-Verlag New York, Inc. (1997)
24. Rudolph, S.: Exploring Relational Structures via $\mathcal{FLE}$. In: Proceedings of 12th International Conference on Conceptual Structures (2004)