PAC Learning of Concept Inclusions for Ontology-Mediated Query Answering (Extended Abstract)

Sergei Obiedkov[1] , **Barış Sertkaya**[2]

[1]Faculty of Computer Science / cfaed / ScaDS.AI, TU Dresden, Dresden, Germany
[2]Frankfurt University of Applied Sciences, Frankfurt, Germany
sergei.obiedkov@tu-dresden.de, sertkaya@fb2.fra-uas.de

1 Introduction

We propose a practical method for learning axioms in a Description Logic (DL) ontology using techniques from probably approximately correct (PAC) learning. The goal is to support ontology-mediated query answering (OMQA) (Bienvenu 2016) by approximating an unknown TBox $\mathcal{T}$ through interaction with a *domain expert oracle* that can decide whether a concept inclusion (CI) is entailed by $\mathcal{T}$. Such an oracle may be instantiated in different ways, e.g., as a human domain expert; a large language model (LLM); a dataset representative of the domain; or a large, complex ontology from which a smaller, focused one is to be distilled.

Our method learns subsumption relationships among conjunctions over a finite *base set* $\mathcal{C}$ of concept descriptions. This set constrains the the search space of candidate axioms and can be tailored to the application, e.g., by restricting concept depth or selecting domain-relevant expressions. We do not fix a particular DL; our results apply to any DL supporting conjunction.

The algorithm employs a *sampling oracle* generating CIs over $\mathcal{C}$ according to an arbitrary fixed distribution $\mathcal{D}$. Given $\epsilon, \delta \in (0, 1)$, it runs in time polynomial in the relevant parameters and returns a TBox $\mathcal{T}'$ such that, with probability at least $1 - \delta$ (over the algorithm's random choices), the probability (under $\mathcal{D}$) that a CI over $\mathcal{C}$ is entailed by exactly one of $\mathcal{T}$ and $\mathcal{T}'$ is at most ϵ.

We also show how to direct the learning process toward subsumptions relevant to a given ABox $\mathcal{A}$, by adapting the distribution $\mathcal{D}$. This enables the learned axioms to improve recall in query answering over incomplete datasets.

2 PAC Learning of Concept Inclusions

Let $\mathcal{T}$ be a TBox, let $\mathcal{C}$ be a *base set* of concept descriptions over its signature, and let $\mathcal{D}$ be a probability distribution on CIs of the form $\bigsqcap \mathcal{X} \sqsubseteq D$, where $\mathcal{X} \subseteq \mathcal{C}$ and $D \in \mathcal{C}$. For $0 < \epsilon < 1$, we say that a set $\mathcal{T}'$ of such CIs is an *ϵ-$\mathcal{C}$-approximation* of $\mathcal{T}$ if $\Pr_{\mathcal{D}}(q \mid (\mathcal{T} \models q) \iff (\mathcal{T}' \not\models q)) \leq \epsilon$, where q is such a CI. $\mathcal{T}'$ is called a *lower ϵ-$\mathcal{C}$-approximation* if $\mathcal{T} \models \mathcal{T}'$.

We base our solution on an algorithm for exactly learning propositional Horn formulas (Angluin, Frazier, and Pitt 1992), which requires two types of queries: membership and equivalence queries. We simulate membership queries using subsumption queries. We modify the equivalence oracle so that, instead of returning a model of exactly one of the two non-equivalent Horn formulas, it returns the GCI corresponding to a Horn clause entailed by exactly one of the two formulas. A PAC algorithm is obtained by replacing each call to this equivalence oracle with an appropriate number of calls to a suitable sampling oracle.

3 Varying the Query Distribution

Our notion of approximation is based on a distribution $\mathcal{D}$ over subsumption queries, intended to capture user-relevant reasoning tasks. In a basic scenario, users explicitly pose such queries and $\mathcal{D}$ reflects their frequency.

A particularly relevant application is ontology-mediated query answering (Bienvenu and Ortiz 2015; Bienvenu 2016). While this setting is typically studied under the assumption of a fully given TBox, in practice this assumption may not hold, e.g., when ontologies are incomplete, evolving, or only partially specified in data-driven or integrated systems. This motivates learning-based approaches that approximate the missing TBox from interaction or data.

Given a query q and a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, OMQA computes all instances of q in $\mathcal{K}$. When $\mathcal{T}$ is incomplete or unavailable, our PAC learning approach can be used to approximate it through interaction with an expert or from a representative dataset, yielding a model that aims to preserve query answering quality in terms of precision and recall.

Definition 1. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, $\mathcal{T}'$ be a TBox, and q be a query. Using certain-answer semantics (Bienvenu and Ortiz 2015), we define* $\operatorname{cert}(q, \mathcal{K})$ *as the set of individual names a from $\mathcal{A}$ satisfying $\mathcal{K} \models q(a)$. The* precision *and* recall *of $\mathcal{T}'$ for q on $\mathcal{K}$ are, respectively,*

$$\mathrm{P}^q(\mathcal{T}', \mathcal{K}) = \frac{|\operatorname{cert}(q, (\mathcal{T}', \mathcal{A})) \cap \operatorname{cert}(q, \mathcal{K})|}{|\operatorname{cert}(q, (\mathcal{T}', \mathcal{A}))|},$$

$$\mathrm{R}^q(\mathcal{T}', \mathcal{K}) = \frac{|\operatorname{cert}(q, (\mathcal{T}', \mathcal{A})) \cap \operatorname{cert}(q, \mathcal{K})|}{|\operatorname{cert}(q, \mathcal{K})|}.$$

If the denominator is 0, then the value of the corresponding measure is defined to be 1.

There are two standard ways to aggregate precision and recall for several queries: macroaveraging and microaveraging (Manning, Raghavan, and Schütze 2008).

Definition 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, $\mathcal{T}'$ be a TBox, and Q be a finite set of queries. The* macro precision *and* recall *of $\mathcal{T}'$ for Q on $\mathcal{K}$ are the average values of the precision and recall over all queries from Q:*

$$\mathrm{P}^{Q}_{\mathrm{macro}}(\mathcal{T}', \mathcal{K}) = \frac{\sum_{q \in Q} \mathrm{P}^{q}(\mathcal{T}', \mathcal{K})}{|Q|},$$

$$\mathrm{R}^{Q}_{\mathrm{macro}}(\mathcal{T}', \mathcal{K}) = \frac{\sum_{q \in Q} \mathrm{R}^{q}(\mathcal{T}', \mathcal{K})}{|Q|}.$$

The micro precision $\mathrm{P}^{Q}_{\mathrm{micro}}(\mathcal{T}', \mathcal{K})$ *and* micro recall $\mathrm{R}^{Q}_{\mathrm{micro}}(\mathcal{T}', \mathcal{K})$ *are defined, respectively, as*

$$\frac{\sum_{q \in Q} |\operatorname{cert}(q, (\mathcal{T}', \mathcal{A})) \cap \operatorname{cert}(q, \mathcal{K})|}{\sum_{q \in Q} |\operatorname{cert}(q, (\mathcal{T}', \mathcal{A}))|},$$

$$\frac{\sum_{q \in Q} |\operatorname{cert}(q, (\mathcal{T}', \mathcal{A})) \cap \operatorname{cert}(q, \mathcal{K})|}{\sum_{q \in Q} |\operatorname{cert}(q, \mathcal{K})|}.$$

The goal in our OMQA scenario is to learn an approximation $\mathcal{T}'$ of $\mathcal{T}$ with high values of the macro/micro precision and recall for some set Q of queries. If $\mathcal{T}'$ is a lower approximation of $\mathcal{T}$, then the precision for every query is 1, and so are the macro and micro precision. In this case, we aim to maximize the recall. Next we describe a heuristic approach to choosing the distribution of subsumption queries in the learning algorithm so as to increase the micro recall on a given ABox $\mathcal{A}$.

Consider a subsumption query $\bigsqcap \mathcal{X} \sqsubseteq D$. If we care about micro recall, it seems important to pose this query whenever $\bigsqcap \mathcal{X}$ has a lot of instances in $\mathcal{K}_0 = (\emptyset, \mathcal{A})$, since a positive answer to the query would then allow us to correctly assert $D(x)$ for many individuals x. Therefore, it seems reasonable to generate the left-hand sides $\bigsqcap \mathcal{X}$ of subsumption queries proportionally to $|\operatorname{cert}(\bigsqcap \mathcal{X}, \mathcal{K})|$. Regarding the right-hand sides, if $D(x)$ rarely occurs in $\mathcal{A}$, this may be due to two reasons: D is a rare concept, or D is a generalization of other concepts and $D(x)$ can be inferred from the target TBox $\mathcal{T}$ together with what is explicitly asserted in $\mathcal{A}$ about x. We cannot tell which of the two it is; so we may want to assume the second case to be on the safe side. Then, we may want to generate the right-hand sides D of subsumption queries with probabilities proportional to $|\operatorname{Ind}(\mathcal{A}) \setminus \operatorname{cert}(D, \mathcal{K})|$, i.e., to the number of individuals that are not (yet) known to be instances of D.

A problem with this approach is that it does not allow us to learn $B \sqsubseteq C$ if B has no instances in $\mathcal{K}$. To address this, we need to change the distribution on the fly, so as to take into account what has already been learned. Thus, having learned $A \sqsubseteq B$, we update $\mathcal{K}$ by replacing $\mathcal{T}_0 = \emptyset$ with $\mathcal{T}_1 = \{A \sqsubseteq B\}$ and recalculate the probabilities involved in sampling premises with respect to $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A})$. Now, $|\operatorname{cert}(B, \mathcal{K}_1)| > 0$, which makes it possible to learn $B \sqsubseteq C$.

This was the method used in the experiments presented in (Obiedkov and Sertkaya 2025). However, it prioritizes concepts $\bigsqcap \mathcal{X}$ with a large number of instances in $\mathcal{A}$ even when these instances are the same for many different $\mathcal{X}$. This can negatively affect precision or recall for certain concepts in $\mathcal{C}$. Instead, when sampling left-hand sides of CIs, we aim to maximize the coverage of individuals in $\mathcal{A}$. Therefore, in further experiments, we adopt a two-stage approach: first, sample an individual a from $\mathcal{A}$ uniformly at random; then sample a subset of $\{C \in \mathcal{C} \mid a \in \operatorname{cert}(C, \mathcal{K}_i)\}$, also uniformly at random.

4 Experimental Evaluation

We implemented our approach in a prototype tool, PACLO[1], and conducted an experimental evaluation in the OMQA context using test ontologies from the repository of the OWL Reasoner Evaluation Workshop (ORE 2015). The expert was simulated using the target TBox $\mathcal{T}$; that is, a subsumption query q is answered positively if and only if $\mathcal{T} \models q$. Subsumption queries were answered using the ELK reasoner (Kazakov, Krötzsch, and Simancik 2014). We set $\delta = 0.001$ and varied ϵ over the values $0.005, 0.01$, and 0.1. Each setting is defined by a base set, an approximation type (ϵ- or lower ϵ-approximation), and a query distribution (uniform or $\mathcal{A}$-induced, as described above).

The test results, presented in (Obiedkov and Sertkaya 2025), show that $\mathcal{A}$-induced distributions typically yield substantially higher recall, especially micro recall, while keeping the number of GCIs small. For lower approximations, the gain in recall is slightly smaller, but they may be preferable when perfect precision is required. For the uniform distribution, lower approximations show some improvement over the results with the empty TBox but remain inferior to those obtained with the $\mathcal{A}$-induced distribution and typically require larger sets of GCIs.

Acknowledgements

Partly supported by DFG in project 389792660 (TRR 248, Center for Perspicuous Systems), by BMBF in ScaDS.AI, and by BMBF and DAAD in project 57616814 (SECAI).

References

Angluin, D.; Frazier, M.; and Pitt, L. 1992. Learning conjunctions of horn clauses. *Machine Learning* 9:147–164.

Bienvenu, M., and Ortiz, M. 2015. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web. 11th Int. S. School, 2015*, LNCS. Springer.

Bienvenu, M. 2016. Ontology-mediated query answering: Harnessing knowledge to get more from data. In *Proc. of IJCAI 2016*. IJCAI/AAAI Press.

Kazakov, Y.; Krötzsch, M.; and Simancik, F. 2014. The incredible ELK - from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reason.* 53(1).

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge Uni. Press.

Obiedkov, S., and Sertkaya, B. 2025. PAC learning of concept inclusions for ontology-mediated query answering. *International Journal of Approximate Reasoning* 186:109523.

[1] https://github.com/sertkaya/paclo