

Learning Query Inseparable \mathcal{ELH} Ontologies

Ana Ozaki, Cosimo Persia, Andrea Mazzullo

KRDB Research Centre, Free University of Bozen-Bolzano

Abstract

We investigate the complexity of learning query inseparable \mathcal{ELH} ontologies in a variant of Angluin’s exact learning model. Given a fixed data instance \mathcal{A}_* and a query language \mathcal{Q} , we are interested in computing an ontology \mathcal{H} that entails the same queries as a target ontology \mathcal{T} on \mathcal{A}_* , that is, \mathcal{H} and \mathcal{T} are inseparable w.r.t. \mathcal{A}_* and \mathcal{Q} . The learner is allowed to pose two kinds of questions. The first is ‘Does $(\mathcal{T}, \mathcal{A}) \models q$?’ with \mathcal{A} an arbitrary data instance and q and query in \mathcal{Q} . An oracle replies this question with ‘yes’ or ‘no’. In the second, the learner asks ‘Are \mathcal{H} and \mathcal{T} inseparable w.r.t. \mathcal{A}_* and \mathcal{Q} ?’. If so, the learning process finishes, otherwise, the learner receives (\mathcal{A}_*, q) with $q \in \mathcal{Q}$, $(\mathcal{T}, \mathcal{A}_*) \models q$ and $(\mathcal{H}, \mathcal{A}_*) \not\models q$ (or vice-versa). Then, we analyse conditions in which query inseparability is preserved if \mathcal{A}_* changes. Finally, we consider the PAC learning model and a setting where the algorithms learn from a batch of classified data, limiting interactions with the oracles.

Introduction

Ontologies are a formal and popular way of representing knowledge. Taxonomies, categorisation of websites, products and their features, as well as more complex and specialized domain knowledge, can be represented with ontologies. Domain experts use ontologies while sharing and annotating information in their fields because in this way knowledge can be unambiguously understood and easily distributed. Medicine, for example, has produced large, standardised, and scalable ontologies (e.g. Galen and SNOMED CT). Broad and general so called knowledge graphs are emerging such as DBPedia (Bizer et al. 2009), Wikidata (Vrandečić and Krötzsch 2014), YAGO (Suchanek, Kasneci, and Weikum 2008). An ontology enables machines to process relations and definitions and reason about that knowledge. Sharing information, formalising a domain, and making assumptions explicit are some of the main reasons for using an ontology.

Designing ontologies is a hard and error-prone task. The research community has approached the problem by developing editors that help ontology engineers to build ontologies manually (Knublauch et al. 2004) and defined design principles (Stuckenschmidt, Parent, and Spaccapietra 2009). Even with tools, building ontologies is a laborious task that also needs expertise. An expert in designing ontologies is called

an ontology engineer. Such an expert is normally familiar with the tools, languages, and techniques necessary to design an ontology through the communication with domain experts. The ontology engineer must communicate and understand the knowledge given by domain experts and then design an ontology that captures what is relevant in the domain.

One of the main challenges in the process of building an ontology is that it often relies on the communication between the ontology engineer (or multiple ontology engineers) and domain experts that, in order to share knowledge, use the ambiguous natural language. Indeed, it can happen that some errors are made while designing it due to the difficulty of sharing knowledge. The ontology engineer can misunderstand the domain experts or they can inadvertently omit precious details. Moreover, knowledge can be implicit and while designing an ontology it is not easy to understand what are the real relationships between concepts and imagine all the possible consequences of designing concepts and their relationships in a particular way. There are several aspects which can influence the difficulty of creating an ontology.

Following the approach by (Konev et al. 2018; Konev, Ozaki, and Wolter 2016), we focus on the problem of finding how concepts should be logically related, assuming that the relevant vocabulary is known by the domain experts, which can share this information with the ontology engineer. In this approach, the problem of building an ontology is treated as a learning problem in which the ontology engineer plays the role of a learner and communicates with the domain experts, who play the role of a teacher (also called an oracle). We assume that (1) the domain experts know the relevant knowledge about the domain and act in a consistent way as a single teacher; (2) the vocabulary that should be used in the ontology is known by the teacher and the learner; (3) the learner can pose queries to the teacher in order to acquire missing knowledge or check if it has learned enough in order to stop learning. The described model can be seen as an instance of Angluin’s exact learning model (Angluin 1988) with membership and equivalence queries. The queries asked by the learner in order to acquire knowledge can be considered as membership queries and the queries that ask if the hypothesis of the learner correctly represents the relevant knowledge of the domain experts can be treated as equivalence queries.

In the exact learning literature, one of the main goals is to determine the complexity of learning an abstract target.

Framework	$\mathcal{EL}(\mathcal{H})_{lhs}$	$\mathcal{EL}(\mathcal{H})_{rhs}$	$\mathcal{EL}(\mathcal{H})$
Equiv.	AQ	✓	–
	IQ	✓	✗
	CQ_r	✓	?
	CQ	✓	✗
Insep.	AQ	✓	✓
	IQ	✓	✓
	CQ_r	✓	✓
	CQ	✓	✗
PAC	AQ	✓	✓
	IQ	✓	✓
	CQ_r	✓	✓
	CQ	✓	✓

Table 1: Polynomial query learnability of learning frameworks.

The complexity depends on the number and size of queries posed by the learner. In the work by (Konev et al. 2018; Konev, Ozaki, and Wolter 2016) it has been shown that exactly learnability of ontologies formulated in the very popular \mathcal{ELH} (Baader et al. 2007a) ontology language is not possible with polynomially many polynomial size queries. Here, we investigate a more flexible setting, where the ontology does not need to be logically equivalent but only inseparable w.r.t. a query language (Lutz and Wolter 2010) and a fixed data instance¹. Query inseparability is the basic requirement for ontology mediated query answering (OMQA) (Bivenvenu 2016). In OMQA, relevant tasks such as ontology versioning, modularisation, update, and forgetting, depend on comparisons between ontologies based on answers given to queries (Botoeva et al. 2019). We study polynomial query and time learnability of the \mathcal{ELH} ontology language in the OMQA setting. The query languages considered are atomic queries (AQ), instance queries (IQ), conjunctive queries (CQ) and a fragment of CQs called rooted CQs (denoted CQ_r). In particular, we show that in our setting the picture is brighter and \mathcal{ELH} can be polynomially learned from IQs, however, it is still not possible to learn this language from CQs.

Table 1 shows the different results obtained by previous works (Konev et al. 2018; Konev, Ozaki, and Wolter 2016) for logical equivalence and our results (shaded in gray) for query inseparability, taking into account the ontology (upper side) and the query languages (left side). ✓ means a positive result, i.e. polynomial query learnability; – means that the query language is not expressive enough for exchanging information and ✗ means that polynomial query learnability cannot be achieved. Polynomial time learnability implies polynomial query learnability but the converse does not hold. All positive results for AQs, IQs, and CQ_r s also hold for polynomial time learnability. Since the learned ontology is not equivalent to the target, it may be the case that after the data is updated the learned ontology and the target are no longer query inseparable. We thus investigate conditions under which query inseparability is preserved when the data changes, which means that no further learning steps are needed after the change. In

¹Here the term ‘query’ refers to queries in the context of databases and query answering. Our data instances are ABoxes.

many application scenarios, interactions with teachers may not be a viable option. We also consider learnability when the learner has only access to a batch of classified examples. Finally, we adapt the Probably Approximately Correct (PAC) model (Valiant 1984) to our OMQA setting, which we separate from the exact and query inseparable problem settings (Theorem 15). Our polynomial time results for query inseparability are transferable to the PAC model extended with membership queries (Theorem 14). Omitted proofs are available at <https://arxiv.org/abs/1911.07229>.

Related Work. To cover the vast literature on ontology learning, we point to the collection edited by (Lehmann and Völker 2014) and surveys authored by (Cimiano, Völker, and Buitelaar 2010) and (Wong, Liu, and Bennamoun 2012). The closest works are the already mentioned papers on exact learning of lightweight description logics (DLs) (Konev et al. 2018; Duarte, Konev, and Ozaki 2018; Konev, Ozaki, and Wolter 2016). Exact learning of concepts formulated in the DL CLASSIC has been investigated by (Cohen and Hirsh 1994) and (Frazier and Pitt 1996). Some other works which are more closely related include works on learning \mathcal{EL} concepts (Funk et al. 2019; Lehmann and Haase 2009). Formal Concept Analysis has been applied for learning DL ontologies (Rudolph 2004; Baader et al. 2007b; Borchmann and Distel 2011; Borchmann 2014; Ganter et al. 2016). Learnability of \mathcal{EL} ontologies from finite interpretations has also been investigated (Klarman and Britz 2015). Association rule mining has been used to learn DL ontologies (with concept expressions of limited depth) (Sazonau and Sattler 2017; Völker and Niepert 2011; Fleischhacker, Völker, and Stuckenschmidt 2012; Völker, Fleischhacker, and Stuckenschmidt 2015).

Basic Definitions

Ontologies and Queries. The \mathcal{ELH} syntax is defined upon mutually disjoint countably infinite sets of concept names N_C , denoted with A, B , role names N_R , denoted with r, s , and individual names N_I , denoted with a, b . \mathcal{EL} -concept expressions C are defined inductively according to the rule $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$, where $A \in N_C$ and $r \in N_R$. For simplicity, we omit \mathcal{EL} - from \mathcal{EL} -concept expressions. An \mathcal{ELH} ontology, also called *TBox*, is a finite set of *concept inclusions* (CI) $C \sqsubseteq D$, where C, D are concept expressions, and *role inclusions* (RI) $r \sqsubseteq s$, where $r, s \in N_R$. We call an \mathcal{ELH} TBox \mathcal{T} a *terminology* if for all $C \sqsubseteq D \in \mathcal{T}$ either C or D is a concept name² and \mathcal{T} has at most one³ inclusion of the form $A \sqsubseteq C$ for every $A \in N_C$. From now on we assume all \mathcal{ELH} TBoxes we deal with are terminologies. An *ABox* \mathcal{A} is a finite set of expressions of the form $A(a)$ or $r(a, b)$, called *assertions*, with $A \in N_C$, $r \in N_R$ and $a, b \in N_I$. We denote by $\text{ind}(\mathcal{A})$ the set of individual names occurring in an

²In the literature, the term *terminology* commonly refers to sets of concept inclusions $A \sqsubseteq C$ and concept definitions $A \equiv C$, with no concept name occurring more than once on the left. As $A \equiv C$ can be equivalently rewritten as $A \sqsubseteq C$ and $C \sqsubseteq A$, our definition is a natural extension of this one.

³If a terminology contains $A \sqsubseteq C$ and $A \sqsubseteq D$ one can always rewrite it into $A \sqsubseteq C \sqcap D$.

ABox \mathcal{A} . The *signature* $\Sigma_{\mathcal{T}}$ of a TBox \mathcal{T} is the set of concept and role names occurring in it, and similarly for the signature $\Sigma_{\mathcal{A}}$ of an ABox \mathcal{A} . A *knowledge base* (KB) is a pair $(\mathcal{T}, \mathcal{A})$ where \mathcal{T} is a TBox and \mathcal{A} is an ABox. We investigate classical query languages considered in the OMQA literature. An AQ takes the form of an assertion. An IQ is of the form $C(a)$ or $r(a, b)$, where C is a concept expression, $r \in \mathbb{N}_R$ and $a, b \in \mathbb{N}_I$. A CQ is a first-order sentence $\exists \vec{x} \varphi(\vec{a}, \vec{x})$, where φ is a conjunction of atoms of the form $r(t_1, t_2)$ or $A(t)$, where t_1, t_2, t (called *terms*) can be individual names from \vec{a} or individual variables from \vec{x} . With an abuse of notation, we denote by AQ, IQ and CQ the sets of atomic, instance and conjunctive queries q , respectively, and we call a *query language* a set $\mathcal{Q} \in \{\text{AQ, IQ, CQ}\}$. The *size* of a concept expression C (TBox \mathcal{T} , ABox \mathcal{A} , query q), denoted by $|C|$ (and, respectively, $|\mathcal{T}|, |\mathcal{A}|, |q|$) is the length of the string that represents it, where concept, role, and individual names are considered to be of length one.

The semantics of \mathcal{ELH} is given as follows. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, called *domain*, and $\cdot^{\mathcal{I}}$ is a function that maps every $a \in \mathbb{N}_I$ to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every $A \in \mathbb{N}_C$ to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and every $r \in \mathbb{N}_R$ to $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ extends to other concept expressions as follows: $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{there is } e \in C^{\mathcal{I}} : (d, e) \in r^{\mathcal{I}}\}$. An interpretation \mathcal{I} *satisfies* a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and an RI $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. It satisfies an assertion $A(a)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and an assertion $r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. \mathcal{I} satisfies an AQ if it satisfies the corresponding assertion, and it satisfies an IQ $C(a)$, or $r(a, b)$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, or $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. \mathcal{I} satisfies a CQ q (or there is a homomorphism from q to \mathcal{I}) if there is a function π , mapping terms of q to elements of $\Delta^{\mathcal{I}}$, such that: $\pi(t) = t^{\mathcal{I}}$, if $t \in \mathbb{N}_I$; $\pi(t) \in A^{\mathcal{I}}$, for every $A(t)$ of q ; and $(\pi(t_1), \pi(t_2)) \in r^{\mathcal{I}}$, for every $r(t_1, t_2)$ of q . We write $\mathcal{I} \models \alpha$ to state that \mathcal{I} satisfies a CI, RI, assertion, or query α . \mathcal{I} satisfies a TBox \mathcal{T} , if it is a model of every CI and RI in \mathcal{T} , and it satisfies an ABox \mathcal{A} if it satisfies every assertion in \mathcal{A} . \mathcal{I} satisfies a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, written $\mathcal{I} \models \mathcal{K}$, if it satisfies both \mathcal{T} and \mathcal{A} . A KB \mathcal{K} *entails* a CI, an RI, an assertion, or query α , written $\mathcal{K} \models \alpha$, if, for all interpretations \mathcal{I} , $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models \alpha$. A KB \mathcal{K} entails a KB \mathcal{K}' , written $\mathcal{K} \models \mathcal{K}'$, if $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models \mathcal{K}'$; \mathcal{K} and \mathcal{K}' are *equivalent*, in symbols $\mathcal{K} \equiv \mathcal{K}'$, if $\mathcal{K} \models \mathcal{K}'$ and $\mathcal{K}' \models \mathcal{K}$. We may also speak of entailments and equivalences of TBoxes and ABoxes, defined as usual (Baader et al. 2007a). For $\mathcal{Q} \in \{\text{AQ, IQ, CQ}\}$, the KBs \mathcal{K} and \mathcal{K}' are *\mathcal{Q} -inseparable*, in symbols $\mathcal{K} \equiv_{\mathcal{Q}} \mathcal{K}'$, if for every query $q \in \mathcal{Q}$, we have that $\mathcal{K} \models q$ iff $\mathcal{K}' \models q$.

Tree Representation and Homomorphisms. We will also represent a concept expression C as a finite directed tree $T_C = (\mathfrak{V}_C, \mathfrak{E}_C, l_C)$, where \mathfrak{V}_C is the set of all vertices, with the root denoted by ρ_C , \mathfrak{E}_C is the set of all edges, and l_C is a labelling function that maps every node to a set of concept names and every edge to a role name. This *tree representation* of C uniquely represents the corresponding concept expression, and it is inductively defined as follows:

- for $C = \top$, $\mathfrak{V}_C = \{\rho_C\}$ and $l_C(\rho_C) = \emptyset$;
- for $C = A$, where $A \in \mathbb{N}_C$, $\mathfrak{V}_C = \{\rho_C\}$ and $l_C(\rho_C) = A$;
- for $C = \exists r.D$, T_C is obtained from T_D by adding a new

root ρ_C and an edge from ρ_C to the root ρ_D of T_D with label $l_C(\rho_C, \rho_D) = r$ (we call ρ_D an *r -successor* of ρ_C);

- for $C = D_1 \sqcap D_2$, T_C is obtained by identifying the roots of T_{D_1} and T_{D_2} , with $l_C(\rho_C) = l_{D_1}(\rho_{D_1}) \cup l_{D_2}(\rho_{D_2})$.

The *ABox representation* of C , \mathcal{A}_C , is the ABox encoding the tree representation T_C of C , defined as follows. For each $v \in \mathfrak{V}_C$, we associate $a_v \in \mathbb{N}_I$. Then, for every $u \in \mathfrak{V}_C$ and every $(u, v) \in \mathfrak{E}_C$, we put: $A(a_u) \in \mathcal{A}_C$ iff $l_C(u) = A$; $r(a_u, a_v) \in \mathcal{A}_C$ iff $l_C(u, v) = r$. Given $\mathcal{A}, \mathcal{A}'$ be ABoxes, a function $h: \text{ind}(\mathcal{A}) \rightarrow \text{ind}(\mathcal{A}')$ is called an *ABox homomorphism* from \mathcal{A} to \mathcal{A}' if: for every $C(a) \in \mathcal{A}$, $C(h(a)) \in \mathcal{A}'$; and, for every $r(a, b) \in \mathcal{A}$, $r(h(a), h(b)) \in \mathcal{A}'$.

Learning Model. We provide basic notions related to the exact learning model, extending the notation in (Konev et al. 2018). A *learning framework* \mathfrak{F} is a quadruple $(\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, where \mathcal{E} is a set of *examples*, \mathcal{S} is a subset of \mathcal{E} , \mathcal{L} is a set of *concept representations* (also called *hypothesis space*), and μ is a mapping from \mathcal{L} to $2^{\mathcal{E}}$. We omit \mathcal{S} if $\mathcal{S} = \mathcal{E}$. Each element l of \mathcal{L} is assumed to be represented using a set of symbols Σ_l (if l is a TBox \mathcal{T} , $\Sigma_{\mathcal{T}}$ is the signature of \mathcal{T}). We say that $e \in \mathcal{E}$ is a *positive example* for $l \in \mathcal{L}$ if $e \in \mu(l)$ and a *negative example* for l if $e \notin \mu(l)$.

Given a learning framework $\mathfrak{F} = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, we are interested in the exact identification of a *target* concept representation $t \in \mathcal{L}$ w.r.t. the subset \mathcal{S} of examples, by posing queries to oracles. Let $\text{MQ}_{\mathfrak{F}, t}$ be the oracle that takes as input some $e \in \mathcal{E}$ and returns ‘yes’ if $e \in \mu(t)$ and ‘no’ otherwise. A *membership query* is a call to the oracle $\text{MQ}_{\mathfrak{F}, t}$. For every $t \in \mathcal{L}$, we denote by $\text{EQ}_{\mathfrak{F}, t}$ the oracle that takes as input a *hypothesis* concept representation $h \in \mathcal{L}$ and returns ‘yes’ if $\mu(h) \cap \mathcal{S} = \mu(t) \cap \mathcal{S}$ and a *counterexample* $e \in (\mu(h) \oplus \mu(t)) \cap \mathcal{S}$ otherwise, where \oplus denotes the symmetric set difference. There is no assumption regarding which counterexample in $(\mu(h) \oplus \mu(t)) \cap \mathcal{S}$ is chosen by the oracle. An *equivalence query with respect to \mathcal{S}* is a call to the oracle $\text{EQ}_{\mathfrak{F}, t}$ (if $\mathcal{S} = \mathcal{E}$, we omit ‘with respect to \mathcal{S} ’).

An (*exact*) *learning algorithm* for $\mathfrak{F} = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ is a deterministic algorithm that, for a fixed but arbitrary $t \in \mathcal{L}$, takes Σ_t as input, is allowed to make queries to $\text{MQ}_{\mathfrak{F}, t}$ and $\text{EQ}_{\mathfrak{F}, t}$ (without knowing what the target t to be learned is), and that eventually halts and outputs some $h \in \mathcal{L}$ with $\mu(h) \cap \mathcal{S} = \mu(t) \cap \mathcal{S}$. We say that the learning algorithm is *positive bounded* if, in addition, $\mu(h) \subseteq \mu(t)$. We say that \mathfrak{F} is *exact learnable* if there is a learning algorithm for \mathfrak{F} and that \mathfrak{F} is *polynomial query learnable* if it is exact learnable by an algorithm A such that at every step the sum of the sizes of the inputs to membership and equivalence queries made by A up to that step is bounded by a polynomial $p(|t|, |e|)$, where t is the target and $e \in \mathcal{S}$ is the largest counterexample seen so far (Arias 2004). Similarly, \mathfrak{F} is *polynomial time learnable* if it is exact learnable by an algorithm A such that at every step (we count each call to an oracle as one step of computation) of computation the time used by A up to that step is bounded by a polynomial $p(|t|, |e|)$, where $t \in \mathcal{L}$ is the target and $e \in \mathcal{S}$ is the largest counterexample seen so far. We denote by PQUERYL and PTIMEL the class of learning frameworks which are, respectively, polynomial query and polynomial time learnable. Clearly, PTIMEL \subseteq PQUERYL.

We now introduce the special case of learning frameworks which we focus in this work, called *OMQA learning frameworks*. Let \mathcal{L} , \mathcal{A}_* , and \mathcal{Q} be, respectively, an ontology language, a fixed but arbitrary ABox, and a query language. An OMQA learning framework $\mathfrak{F}(\mathcal{L}, \mathcal{A}_*, \mathcal{Q})$ is a learning framework $(\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ where \mathcal{E} is the set of all pairs (\mathcal{A}, q) with \mathcal{A} an ABox (may be different from \mathcal{A}_*) and $q \in \mathcal{Q}$; \mathcal{L} is the set of all TBoxes formulated in \mathcal{L} sharing a common finite signature (we write $\Sigma_{\mathcal{T}}$ to refer to the signature of the target, which is assumed to be same as the one used for the hypothesis); \mathcal{S} is the set of elements (\mathcal{A}, q) of \mathcal{E} where $\mathcal{A} = \mathcal{A}_*$; and, for all $\mathcal{T} \in \mathcal{L}$, $\mu(\mathcal{T}) = \{(\mathcal{A}, q) \in \mathcal{E} \mid (\mathcal{T}, \mathcal{A}) \models q\}$. We assume that the signature of q , i.e., the set of concept and role names occurring in q , is $\Sigma_{\mathcal{T}} \cup \Sigma_{\mathcal{A}}$. Moreover, we define the *size* of an example (\mathcal{A}, q) , denoted by $|(A, q)|$, as the sum of the size of \mathcal{A} and q . Given an OMQA learning framework $\mathfrak{F}(\mathcal{L}, \mathcal{A}_*, \mathcal{Q}) = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, for all $\mathcal{H}, \mathcal{T} \in \mathcal{L}$, if $\mu(\mathcal{H}) \cap \mathcal{S} = \mu(\mathcal{T}) \cap \mathcal{S}$, then \mathcal{H}, \mathcal{T} are \mathcal{Q} -inseparable w.r.t. \mathcal{A}_* . Since an equivalence query in an OMQA learning framework with $\mathcal{S} \neq \mathcal{E}$ is in fact asking whether a given TBox is *query inseparable* from the target TBox w.r.t. a fixed ABox and a query language, we may call it an *inseparability query*.

Polynomial Learnability

We investigate whether, for a given fixed ABox and query language, the problem of learning query inseparable \mathcal{ELH} TBoxes is polynomial. We first discuss the relationship between our OMQA setting and the data retrieval one (Konev, Ozaki, and Wolter 2016). The difference between the two settings is that here the oracle can only choose counterexamples of the form (\mathcal{A}_*, q) , with the fixed ABox \mathcal{A}_* given as input, whereas in the mentioned work the ABox in a counterexample can be arbitrary. In both settings the learner can pose membership queries with an arbitrary ABox in the examples.

Formally, given an ontology language \mathcal{L} and a query language \mathcal{Q} , we denote by $\mathfrak{F}(\mathcal{L}, \mathcal{Q})$ the learning framework $(\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, where \mathcal{E} is the set of all pairs (\mathcal{A}, q) with \mathcal{A} an ABox and $q \in \mathcal{Q}$; \mathcal{L} is the set of all TBoxes formulated in \mathcal{L} ; $\mathcal{E} = \mathcal{S}$; and, for all $\mathcal{T} \in \mathcal{L}$, $\mu(\mathcal{T}) = \{(\mathcal{A}, q) \in \mathcal{E} \mid (\mathcal{T}, \mathcal{A}) \models q\}$. We denote by $\mathcal{ELH}_{\text{lhs}}$ and $\mathcal{ELH}_{\text{rhs}}$ the fragments of \mathcal{ELH} which only allow complex concept expressions on the left-hand side and on the right-hand side of CIs, respectively. It is known that the learning frameworks $\mathfrak{F}(\mathcal{ELH}_{\text{lhs}}, \text{AQ})$ and $\mathfrak{F}(\mathcal{ELH}_{\text{rhs}}, \text{IQ})$ are in PTIMEL, whereas $\mathfrak{F}(\mathcal{ELH}, \text{IQ})$ and $\mathfrak{F}(\mathcal{ELH}_{\text{rhs}}, \text{CQ})$ are not in PQUERYL (Konev, Ozaki, and Wolter 2016). It follows from our definitions that, for any ontology language \mathcal{L} , query language \mathcal{Q} and ABox \mathcal{A}_* , polynomial learnability of $\mathfrak{F}(\mathcal{L}, \mathcal{Q})$ implies polynomial learnability of $\mathfrak{F}(\mathcal{L}, \mathcal{A}_*, \mathcal{Q})$. Thus, the positive results for $\mathfrak{F}(\mathcal{ELH}_{\text{lhs}}, \text{AQ})$ and $\mathfrak{F}(\mathcal{ELH}_{\text{rhs}}, \text{IQ})$ hold in the OMQA setting. That is, for any ABox \mathcal{A}_* , the learning frameworks $\mathfrak{F}(\mathcal{ELH}_{\text{lhs}}, \mathcal{A}_*, \text{AQ})$ and $\mathfrak{F}(\mathcal{ELH}_{\text{rhs}}, \mathcal{A}_*, \text{IQ})$ are in PTIMEL.

Proposition 1. *For any ontology language \mathcal{L} , query language \mathcal{Q} and ABox \mathcal{A}_* , if $\mathfrak{F}(\mathcal{L}, \mathcal{Q})$ is in PTIMEL/PQUERYL, then $\mathfrak{F}(\mathcal{L}, \mathcal{A}_*, \mathcal{Q})$ is in PTIMEL/PQUERYL.*

In the following, we extend the positive results for $\mathcal{ELH}_{\text{lhs}}$ and $\mathcal{ELH}_{\text{rhs}}$ to the class of \mathcal{ELH} terminologies (the union of $\mathcal{ELH}_{\text{lhs}}$ and $\mathcal{ELH}_{\text{rhs}}$) in the OMQA setting based on IQs.

This is in contrast with the negative result for the class of \mathcal{ELH} terminologies in the data retrieval setting with IQs, thus, showing that the converse direction of Proposition 1 does not hold. Throughout this section \mathcal{A}_* is a fixed but arbitrary ABox. We also analyse the learning framework $\mathfrak{F}(\mathcal{ELH}_{\text{rhs}}, \mathcal{A}_*, \text{CQ})$, which is also not in PQUERYL in the OMQA setting.

Learning \mathcal{ELH} ontologies with AQ. We argue that the learning framework $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{AQ})$ is in PTIMEL. There are only polynomially many counterexamples that the oracle can give since they can only be of the form (\mathcal{A}_*, q) with q an atomic query (using symbols from $\Sigma_{\mathcal{T}}$). The set of RIs entailed by \mathcal{T} can be learned in polynomial time by adding to the hypothesis \mathcal{H} all RIs $r \sqsubseteq s$ such that the membership oracle replies ‘yes’ given an example $(\{r(a, b)\}, s(a, b))$ as input. Therefore we only need to show that one can compute concept inclusions that together with \mathcal{A}_* entail precisely the same atomic queries as the target ontology \mathcal{T} on \mathcal{A}_* . The following lemma establishes that indeed one can compute such concept inclusions in polynomial time.

Lemma 2. *Let \mathcal{T} and \mathcal{H} be resp. the target and the hypothesis (of size polynomial in $|\mathcal{T}|$) terminologies. Given a positive counterexample $(\mathcal{A}_*, A(a))$, one can compute in polynomial time in $|\mathcal{A}_*|$ and $|\mathcal{T}|$ a $\text{CIC} \sqsubseteq B$ such that $(\mathcal{T}, \mathcal{A}_*) \models B(b)$, $(\mathcal{H}, \mathcal{A}_*) \not\models B(b)$, and $(\mathcal{H} \cup \{C \sqsubseteq B\}, \mathcal{A}_*) \models B(b)$, for some $b \in \text{ind}(\mathcal{A}_*)$. Moreover, $\mathcal{T} \models C \sqsubseteq B$.*

The main idea for proving Lemma 2 is to transform the structure of \mathcal{A}_* into a tree shaped structure, as in (Konev, Ozaki, and Wolter 2016). However, in the mentioned work only CIs of the form $C \sqsubseteq A$ are considered, while in our case CIs of the form $A \sqsubseteq C$ may also be present. With such CIs the computed tree shaped ABox may be smaller than the original concept expression in \mathcal{T} (as in Example 3).

Example 3. Assume that $\mathcal{T} = \{B \sqsubseteq \exists s.B, \exists r.\exists s.B \sqsubseteq A\}$ and $\mathcal{A} = \{r(a, b), B(b)\}$. We have that $(\mathcal{T}, \mathcal{A}) \models A(a)$ and the concept expression $\exists r.B$ encoded in \mathcal{A} is smaller than the original concept expression $\exists r.\exists s.B$ in \mathcal{T} implying the concept name A . Intuitively, even though there is no homomorphism from $\mathcal{A}_{\exists r.\exists s.B}$ to \mathcal{A} , we have that B ‘abbreviates’ $\exists s.B$, because it is implied by B . \triangleleft

Since there are polynomially many possible counterexamples, our upper bound for AQs follows from Lemma 2.

Theorem 4. *$\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{AQ})$ is in PTIMEL. Moreover, there is a positive bounded learning algorithm for showing such upper bound which only poses membership queries (no inseparability queries are needed).*

Learning \mathcal{ELH} ontologies with IQ. We build on the result for AQs (Theorem 4) and argue that the learning framework $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{IQ})$ is in PTIMEL. By Theorem 4, CIs of the form $C \sqsubseteq A$ which ensure AQ-inseparability w.r.t. \mathcal{A}_* can be learned with membership queries. It remains to show how one can learn CIs of the form $A \sqsubseteq C$, so that \mathcal{H} is IQ-inseparable from \mathcal{T} (w.r.t. \mathcal{A}_*). By Theorem 4, we can assume that one can construct a hypothesis \mathcal{H} such that $\mathcal{T} \models \mathcal{H}$, since there is a positive bounded learning algorithm. The next lemma states that if $\mathcal{T} \models \mathcal{H}$ and $(\mathcal{H}, \mathcal{A}_*) \equiv_{\text{AQ}} (\mathcal{T}, \mathcal{A}_*)$ then one can transform a counterexample of the form $(\mathcal{A}_*, D(a))$ into

a CI such that $\mathcal{T} \models A \sqsubseteq C$, and $\mathcal{H} \not\models A \sqsubseteq C$, with C a subconcept of D and $A \in \Sigma_{\mathcal{T}}$ such that $(\mathcal{H}, \mathcal{A}_*) \models A(b)$ for some $b \in \mathbb{N}_I$. Since $|\Sigma_{\mathcal{T}}|$ is polynomial in $|\mathcal{T}|$ and the number of subconcepts of D is also polynomial in $|D|$. One can find such $A \sqsubseteq C$ in polynomial time w.r.t. $|\mathcal{T}|$ and $|(\mathcal{A}_*, D(a))|$.

Lemma 5. *Let \mathcal{T} and \mathcal{H} be \mathcal{ELH} terminologies and let \mathcal{A}_* be an ABox. Assume $(\mathcal{H}, \mathcal{A}_*) \equiv_{\text{AQ}} (\mathcal{T}, \mathcal{A}_*)$. If $(\mathcal{A}_*, D(a)) \in \mu(\mathcal{T}) \setminus \mu(\mathcal{H})$ then there is $A(b)$ and a subconcept C of D such that $(\mathcal{H}, \mathcal{A}_*) \models A(b)$, $\mathcal{T} \models A \sqsubseteq C$, and $\mathcal{H} \not\models A \sqsubseteq C$.*

Given $A \sqsubseteq C$ such that $\mathcal{T} \models A \sqsubseteq C$, and $\mathcal{H} \not\models A \sqsubseteq C$, one can compute with polynomially many polynomial size queries another CI $A' \sqsubseteq C'$ entailed by \mathcal{T} but not by \mathcal{H} belonging to a class of CIs called \mathcal{T} -essential (Konev et al. 2018, Lemma 29). In fact this can be done in polynomial time given the complexity of entailment checking (Baader, Lutz, and Brandt 2008) (no inverse roles). Such \mathcal{T} -essential CIs have the property that their size is bounded polynomially in the size of \mathcal{T} (Konev et al. 2018, Lemma 32) and if $\alpha_1 = A' \sqsubseteq C_1$ and $\alpha_2 = A' \sqsubseteq C_2$ are \mathcal{T} -essential and not equivalent then one can compute in polynomial time a \mathcal{T} -essential CI $A' \sqsubseteq C'$ such that it entails α_1 and α_2 (Konev et al. 2018, Lemma 30). All in all, if the learner computes such \mathcal{T} -essential counterexamples and adds/refines them in the hypothesis (see (Konev et al. 2018, Algorithm 2)) then, after learning from polynomially many counterexamples, it will terminate and output a hypothesis IQ-inseparable from the target (w.r.t. \mathcal{A}_*). The presence of CIs of the form $C \sqsubseteq A$ does not affect this result (Duarte, Konev, and Ozaki 2018).

Theorem 6. $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{IQ})$ is in PTIMEL.

In contrast, the learning framework $\mathfrak{F}(\mathcal{ELH}, \text{IQ})$ is not in PQUERYL (Konev, Ozaki, and Wolter 2016). We observe that the counterexamples used in such hardness proof are based on an exponential number of ABoxes encoding concept expressions of the form $C_{\mathbf{b}} = \prod_{i \leq n} C_i$, where $\mathbf{b} = b_1 \dots b_n$ is a sequence with $b_i \in \{0, 1\}$, $C_i = A_i$ if $b_i = 1$, and $C_i = B_i$ if $b_i = 0$. In the OMQA setting, the ABox is fixed and, as stated in Theorem 6, this lowers the complexity.

Learning \mathcal{ELH} ontologies with CQ. \mathcal{ELH} ontologies are not polynomial query learnable in the data retrieval setting with CQs as the query language (in fact not even the fragment $\mathcal{ELH}_{\text{rhs}}$) (Konev, Ozaki, and Wolter 2016). The counterexamples used by the oracle in the hardness proof are of the form $(\{A(a)\}, q)$ (Konev, Ozaki, and Wolter 2016, proof of Lemma 8), so $\{A(a)\}$ can be considered as the fixed ABox given as part of the input in an OMQA learning framework. Thus, the mentioned hardness result can be transferred to our setting. We formalise this result with the next theorem.

Theorem 7. $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{CQ})$ is not in PQUERYL.

The hardness proof in the mentioned paper uses a very simple CQ of the form $\exists x M(x)$, which has a match in the anonymous part of the model but ‘hides’ the concept on the right side of a CI that causes the entailment of this query. This phenomenon makes one wonder whether restricting to the class of queries in which every variable needs to be reachable by an individual name (as it happens with IQs) can tame the complexity of the problem. Our next theorem proves this.

Given a CQ q , we define G_q as the directed graph (V, E) where the nodes V are the terms of q and the edges E are the

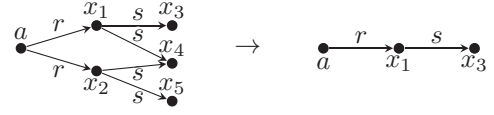


Figure 1: Assume $\mathcal{T} = \{A \sqsubseteq \exists r. \exists s. \top\}$, $\mathcal{A}_* = \{A(a)\}$ and $\mathcal{H} = \emptyset$. A call to $\text{EQ}_{\mathfrak{F}, \mathcal{T}}$ can output $(\mathcal{A}_*, \exists \vec{x} (r(a, x_1) \wedge r(a, x_2) \wedge s(x_1, x_3) \wedge s(x_1, x_4) \wedge s(x_2, x_4) \wedge s(x_2, x_5)))$, which can be converted into $(\mathcal{A}_*, \exists r. \exists s. \top(a))$ by merging variables and asking $\text{MQ}_{\mathfrak{F}, \mathcal{T}}$ whether the new query holds.

pairs (t_1, t_2) such that there is an atom of the form $r(t_1, t_2)$ in q . We say that a CQ $q = \exists \vec{x} \varphi(\vec{a}, \vec{x})$ is *rooted* if for every x in \vec{x} , we have that x is reachable from a node in G_q that is in \vec{a} . We denote by CQ_r the class of all rooted CQs. The next lemma establishes that one can transform queries in CQ_r into queries in IQ (by posing membership queries).

Lemma 8. *Let \mathcal{T} and \mathcal{H} be \mathcal{ELH} TBoxes and assume \mathcal{T} and \mathcal{H} entail the same RIs. Given a positive counterexample (\mathcal{A}_*, q) (for \mathcal{T} and \mathcal{H}), with $q \in \text{CQ}_r$, one can construct a positive counterexample (\mathcal{A}_*, q') with $q' \in \text{IQ}$ in polynomial time in $|(\mathcal{A}_*, q)| |\Sigma_{\mathcal{T}}|$.*

In Figure 1, it is shown an example of this conversion. Even though the conversion involves deciding query answering, which is NP-hard, these checks are on the ‘side’ of the oracle, and so, they do not affect the complexity of learning. By Lemma 8 a IQ-inseparable hypothesis can be found by following the same steps of a learning algorithm for IQ after q in (\mathcal{A}_*, q) is converted into an instance query. For \mathcal{ELH} , if TBoxes entail the same RIs then IQ-inseparability implies CQ_r -inseparability. Since RIs can be easily learned with membership queries, we obtain our next theorem.

Theorem 9. $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{CQ}_r)$ is in PTIMEL.

Data Updates

The algorithm presented in the previous section for IQs computes an ontology \mathcal{H} that is IQ-inseparable from the target \mathcal{T} w.r.t. a fixed ABox \mathcal{A}_* . In this section, we first study when IQ-inseparability is preserved, without changes to the hypothesis \mathcal{H} , if \mathcal{A}_* is updated to an ABox \mathcal{A} . Then, given an OMQA learning framework $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*, \text{IQ}) = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, we determine conditions on an updated ABox \mathcal{A} , sufficient to guarantee that a learning framework $(\mathcal{E}, \mathcal{S}', \mathcal{L}, \mu)$ with $\mathcal{S}' \supseteq \mathcal{S}$ is still in PTIMEL.

To characterise when IQ-inseparability is preserved if \mathcal{A}_* is updated to an ABox \mathcal{A} , we use the classical notion of bisimulation. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be two interpretations. A *bisimulation* is a non-empty relation $\mathcal{Z} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ satisfying the following conditions, for all $(d, e) \in \mathcal{Z}$: (1) for all concept names $A \in \mathbb{N}_C$, $d \in A^{\mathcal{I}}$ iff $e \in A^{\mathcal{J}}$; (2) for all role names $r \in \mathbb{N}_R$, if $(d, d') \in r^{\mathcal{I}}$, $d' \in \Delta^{\mathcal{I}}$, then there exists $e' \in \Delta^{\mathcal{J}}$ such that $(e, e') \in r^{\mathcal{J}}$ and $(d', e') \in \mathcal{Z}$; (3) for all role names $r \in \mathbb{N}_R$, if $(e, e') \in r^{\mathcal{J}}$, $e' \in \Delta^{\mathcal{J}}$, then there exists $d' \in \Delta^{\mathcal{I}}$ such that $(d, d') \in r^{\mathcal{I}}$ and $(d', e') \in \mathcal{Z}$. If $(d, e) \in \mathcal{Z}$, we write $(\mathcal{I}, d) \sim (\mathcal{J}, e)$.

Theorem 10. *Let \mathcal{T} and \mathcal{H} be \mathcal{ELH} terminologies entailing the same RIs, and let \mathcal{A}_* and \mathcal{A} be ABoxes. If, for all $b \in$*

$\text{ind}(\mathcal{A})$, there is $a \in \text{ind}(\mathcal{A}_*)$ such that $(\mathcal{I}_{\mathcal{A}_*}, a) \sim (\mathcal{I}_{\mathcal{A}}, b)$, then $(\mathcal{H}, \mathcal{A}_*) \equiv_{\text{IQ}} (\mathcal{T}, \mathcal{A}_*)$ implies $(\mathcal{H}, \mathcal{A}) \equiv_{\text{IQ}} (\mathcal{T}, \mathcal{A})$.

Theorem 10 does not hold if we require the ABoxes \mathcal{A}_* and \mathcal{A} to be *homomorphically equivalent*, i.e., if there are ABox homomorphisms from \mathcal{A}_* to \mathcal{A} and from \mathcal{A} to \mathcal{A}_* .

Example 11. Consider $\mathcal{T} = \{\exists r.A_1 \sqsubseteq B\}$ and $\mathcal{A}_* = \{r(a, b), A_1(b), A_2(b)\}$. The hypothesis $\mathcal{H} = \{\exists r.(A_1 \sqcap A_2) \sqsubseteq B\}$ is IQ-inseparable. However, if $\mathcal{A} = \mathcal{A}_* \cup \{r(a', b'), A_1(b')\}$, then IQ-inseparability is not preserved, even though \mathcal{A}_* and \mathcal{A} are homomorphically equivalent. \triangleleft

The problem here is that the left-hand side of CIs in \mathcal{H} could be biased and too specific for the individuals in \mathcal{A}_* . Indeed, in Example 11, if the more general concept expression $\exists r.A_1$ on the left-side had been learned, then IQ-inseparability would have been preserved after the update. If we allow for modifications to the learned hypothesis \mathcal{H} , we can extend the class of updated ABoxes \mathcal{A} not only to those in which every individual in \mathcal{A} is bisimilar to an individual in \mathcal{A}_* but in which a more relaxed condition is required. It is easy for the learner to make certain kinds of generalisation, for instance, check whether $\mathcal{T} \models \exists r.A_1 \sqsubseteq B$ and add such more general CI to \mathcal{H} . Therefore, the idea is to suitably ‘generalise’ the left-hand side of CIs in the hypothesis \mathcal{H} computed by the learning algorithm.

Generalisation of $C \sqsubseteq A \in \mathcal{H}$ for \mathcal{T} consists of replacing C by the result C' of (1) replacing a concept name B in C with \top or B' such that $\mathcal{T} \models B \sqsubseteq B'$ and $\mathcal{T} \not\models B' \sqsubseteq B$ if $\mathcal{T} \models C' \sqsubseteq A$; or (2) replacing a role name r in C with s such that $\mathcal{T} \models r \sqsubseteq s$ and $\mathcal{T} \not\models s \sqsubseteq r$ if $\mathcal{T} \models C' \sqsubseteq A$. We say that $C \sqsubseteq A \in \mathcal{H}$ is *generalised* for \mathcal{T} if generalization of $C \sqsubseteq A \in \mathcal{H}$ for \mathcal{T} has been exhaustively applied. \mathcal{H} is generalised for \mathcal{T} if all the CIs in it are generalised. We may omit ‘for \mathcal{T} ’ if this is clear from the context.

With the following definitions, we define a class of ABoxes that are guaranteed to preserve IQ-inseparability if the hypothesis is generalised. Given a TBox \mathcal{T} and concept names $A, B \in \Sigma_{\mathcal{T}} \cap \text{N}_{\mathcal{C}}$, we say that there is a *linear derivation* from A to B if $\mathcal{T} \models A \sqsubseteq B$ and for all $B' \in \text{N}_{\mathcal{C}}$ such that $\mathcal{T} \models A \sqsubseteq B'$ we have that $\mathcal{T} \models B' \sqsubseteq B$. Similarly, for $r, s \in \Sigma_{\mathcal{T}} \cap \text{N}_{\mathcal{R}}$, there is a *linear derivation* from r to s if $\mathcal{T} \models r \sqsubseteq s$ and for all $s' \in \text{N}_{\mathcal{R}}$ such that $\mathcal{T} \models r \sqsubseteq s'$ we have that $\mathcal{T} \models s' \sqsubseteq s$. We write $\mathcal{A} < \mathcal{A}'$ if \mathcal{A}' is the result of replacing $A(a) \in \mathcal{A}$ by $B(a)$ and there is a linear derivation from A to B , or if \mathcal{A}' is the result of replacing $r(a, b) \in \mathcal{A}$ by $s(a, b)$ and there is a linear derivation from r to s . We define $\mathbf{g}_{\mathcal{T}}(\mathcal{A}_*)$ as the set of all ABoxes \mathcal{A} such that there is a sequence $\mathcal{A}_1 < \dots < \mathcal{A}_n$ with $\mathcal{A}_1 = \mathcal{A}_*$ and $\mathcal{A}_n = \mathcal{A}$.

The following theorem establishes an our upper bound for learning frameworks extending \mathcal{S} with all the examples of the form (\mathcal{A}, q) , where $\mathcal{A} \in \mathbf{g}_{\mathcal{T}}(\mathcal{A}_*)$ and $q \in \text{IQ}$.

Theorem 12. Let \mathfrak{F} be the learning framework that results from adding all pairs of the form (\mathcal{A}, q) , with $\mathcal{A} \in \mathbf{g}_{\mathcal{T}}(\mathcal{A}_*)$ and $q \in \text{IQ}$, to the set \mathcal{S} in $\mathfrak{F}(\mathcal{E}\mathcal{L}\mathcal{H}, \mathcal{A}_*, \text{IQ}) = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, where $\mathcal{T} \in \mathcal{L}$. Assume $\Sigma_{\mathcal{T}} \subseteq \Sigma_{\mathcal{A}_*}$. Then, \mathfrak{F} is in PTIMEL .

Learning from Data

The existence of oracles that correctly answer to all the queries posed by the learner does not naturally fit those set-

tings in which only a direct access to data is available. In this section, we study how the oracle-based approach presented so far can be modified so to allow access to examples retrieved from data, thus reducing the dependency of our learning model on membership and inseparability queries. Firstly, we consider a finite batch of examples (Arias, Khardon, and Maloberti 2007), to be used as a representative of the entire data pool, and study conditions under which it is guaranteed the existence of such a batch that allows us to learn inseparable ontologies. Then, we analyse how a data-driven approach can be used as a basis for a learning model for DL ontologies based on the well-known PAC learning model, possibly extended with membership queries (Valiant 1984).

Learning from batch. Given an OMQA learning framework $\mathfrak{F}(\mathcal{L}, \mathcal{A}_*, \mathcal{Q}) = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$, a *batch* \mathcal{B} is a *finite* subset of \mathcal{E} . One could ask under which conditions a batch that allows us to construct an ontology $\mathcal{H} \in \mathcal{L}$ which is \mathcal{Q} -inseparable from a target $\mathcal{T} \in \mathcal{L}$ is guaranteed to exist. If no restrictions are imposed on the form of the examples occurring in \mathcal{B} , the answer is trivial for $\mathcal{E}\mathcal{L}\mathcal{H}$ ontologies and IQs. Indeed, for every $C \sqsubseteq D \in \mathcal{T}$, consider the set \mathcal{B} of examples of the form $(\mathcal{A}_C, D(\rho_C))$, obtained by representing the concept C as a labelled tree with root ρ_C and encoded in the ABox \mathcal{A}_C . These examples have the property that, for every $\mathcal{T} \in \mathcal{L}$, $\mathcal{T} \models C \sqsubseteq D$ iff $(\mathcal{T}, \mathcal{A}_C) \models D(\rho_C)$. By setting

$$\mathcal{H} = \{C \sqsubseteq D \mid (\mathcal{A}_C, D(\rho_C)) \in \mathcal{B}\},$$

we obtain that \mathcal{H} is equivalent to (and thus IQ-inseparable, w.r.t. any ABox, from) \mathcal{T} . This construction can be easily extended to $\mathcal{E}\mathcal{L}\mathcal{H}$ by using examples of the form $(\{r(a, b)\}, s(a, b))$. However, instead of allowing the examples retrieved from our data to have no restrictions in their size and in the shape of the ABoxes, it would be more natural to require that these ABoxes contain less information than \mathcal{A}_* , given as a parameter. This intuition can be made precise by imposing that, for each ABox \mathcal{A} in the batch, there is an ABox homomorphism from \mathcal{A} to \mathcal{A}_* and $|\mathcal{A}|$ is polynomial in $|\mathcal{T}|$. Our next theorem states that, under these assumptions, one can construct a \mathcal{Q} -inseparable $\mathcal{E}\mathcal{L}\mathcal{H}$ terminology.

Theorem 13. Let $\mathfrak{F}(\mathcal{E}\mathcal{L}\mathcal{H}, \mathcal{A}_*, \mathcal{Q}) = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ be an OMQA learning framework, with $\mathcal{Q} \in \{\text{AQ}, \text{IQ}, \text{CQ}_r\}$, and let $\mathcal{T} \in \mathcal{L}$ be such that $\Sigma_{\mathcal{T}} \subseteq \Sigma_{\mathcal{A}_*}$. Let $\mathcal{X} \subseteq \mathcal{E}$ be the set of examples (\mathcal{A}, q) such that there is an ABox homomorphism from \mathcal{A} to \mathcal{A}_* . Then, there is a batch $\mathcal{B} \subseteq \mathcal{X}$, polynomial in $|\mathcal{T}|$, and an algorithm such that it takes \mathcal{B} as input, it eventually halts, and returns $\mathcal{H} \in \mathcal{L}$ such that $\mu(\mathcal{H}) \cap \mathcal{S} = \mu(\mathcal{T}) \cap \mathcal{S}$.

PAC learning. Let $\mathfrak{F} = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ be a learning framework. A *probability distribution* \mathcal{D} on \mathcal{S} is a function $\mathcal{D}: 2^{\mathcal{S}} \rightarrow [0, 1] \subset \mathbb{R}$ such that $\mathcal{D}(\bigcup_{i \in I} X_i) = \sum_{i \in I} \mathcal{D}(X_i)$ for mutually disjoint X_i , where I is a countable set of indices, $X_i \subseteq \mathcal{S}$, and $\mathcal{D}(\mathcal{S}) = 1$. Given a target $t \in \mathcal{L}$, let $\text{EX}_{\mathfrak{F}, t}^{\mathcal{D}}$ be the oracle that takes no input, and outputs a *classified example* $(e, \ell_t(e))$, where $e \in \mathcal{S}$ is sampled according to the probability distribution \mathcal{D} , $\ell_t(e) = 1$, if $e \in \mu(t) \cap \mathcal{S}$ (*positive example*), and $\ell_t(e) = 0$, otherwise (*negative example*). An *example query* is a call to the oracle $\text{EX}_{\mathfrak{F}, t}^{\mathcal{D}}$. A *sample* generated by $\text{EX}_{\mathfrak{F}, t}^{\mathcal{D}}$ is a (multi-)set of indexed classified ex-

amples, independently and identically distributed according to \mathcal{D} , sampled by calling $\text{EX}_{\mathfrak{F},t}^{\mathcal{D}}$. A learning framework \mathfrak{F} is *PAC learnable* if there is a function $f : (0, 1)^2 \rightarrow \mathbb{N}$ and a deterministic algorithm such that, for every $\epsilon, \delta \in (0, 1) \subset \mathbb{R}$, every probability distribution \mathcal{D} on \mathcal{S} , and every target $t \in \mathcal{L}$, given a sample of size $m \geq f(\epsilon, \delta)$ generated by $\text{EX}_{\mathfrak{F},t}^{\mathcal{D}}$, the algorithm always halts and outputs $h \in \mathcal{L}$ such that with probability at least $(1 - \delta)$ over the choice of m examples in \mathcal{S} , we have that $\mathcal{D}((\mu(h) \oplus \mu(t)) \cap \mathcal{S}) \leq \epsilon$. If the time used by the algorithm is bounded by a polynomial function $p(|t|, |e|, 1/\epsilon, 1/\delta)$, where e is the largest example in the sample, then we say that \mathfrak{F} is *polynomial time PAC learnable*. If, in addition, the algorithm is allowed to make membership queries (where each call to $\text{MQ}_{\mathfrak{F},t}$ counts as one step of computation), we say that \mathfrak{F} is *polynomial time PAC learnable with membership queries*.

Theorem 14 ((Angluin 1988), (Mohri, Rostamizadeh, and Talwalkar 2012)). *If \mathfrak{F} is in PTIMEL, then \mathfrak{F} is polynomial time PAC learnable with membership queries.*

However, the converse direction of Theorem 14 does not hold (Blum 1994). The argument in the mentioned paper is based on the assumption that one-way functions exist and cannot be easily adapted to serve as a counterexample for OMQA learning frameworks. Our next result, showing that the converse direction of Theorem 14 does not hold in our setting, does *not* rely on cryptographic assumptions, however, it is representation-dependent. Given a sequence $\sigma = \sigma_1 \sigma_2 \dots \sigma_n$, with $\sigma_i \in \{r, s\}$, let the expression $\exists \sigma.C$ stand for $\exists \sigma_1. \exists \sigma_2. \dots \exists \sigma_n.C$ (clearly, there are 2^n expressions of this form). Consider the OMQA learning framework $\mathfrak{F}(L, \mathcal{A}_*, Q)$ where $\mathcal{A}_* = \{A(a_i)\}$; Q is the query language that extends IQs with a CQ of the form $\exists x M(x)$; and L is an ontology language allowing only $\mathcal{ELH}_{\text{rhs}}$ TBoxes of the form $\mathcal{T}_\sigma = \{A \sqsubseteq \exists \sigma.M\} \cup \mathcal{T}_0$ to be expressed, where

$$\begin{aligned} \mathcal{T}_0 = & \{A \sqsubseteq X_0, M \sqsubseteq \exists r.M \sqcap \exists s.M\} \cup \\ & \{X_i \sqsubseteq \exists r.X_{i+1} \sqcap \exists s.X_{i+1} \mid 0 \leq i < n\}. \end{aligned}$$

It can be shown, with an argument similar to the one used in (Konev, Ozaki, and Wolter 2016, proof of Lemma 8) (and Theorem 7 above), that such framework is not polynomial query learnable. However, due to the restrictions on the hypothesis space, it is polynomial time PAC learnable, even without membership queries.

Theorem 15. *There is a polynomial time PAC learnable OMQA learning framework that is not in PQUERYL.*

A learning framework $\mathfrak{F} = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ *shatters* a set of examples $\mathcal{X} \subseteq \mathcal{S}$ if $|\{\mu(h) \cap \mathcal{X} \mid h \in \mathcal{L}\}| = 2^{|\mathcal{X}|}$. The *VC-dimension* (Vapnik 1995) of \mathfrak{F} , denoted $\text{VC}(\mathfrak{F})$, is the maximal size of a set $\mathcal{X} \subseteq \mathcal{S}$ such that \mathfrak{F} shatters \mathcal{X} . If \mathfrak{F} can shatter arbitrarily large sets then \mathfrak{F} has infinite VC-dimension.

Example 16. For $n \in \mathbb{N}$, let

$$\mathcal{A}_*^n = \{r(a_i, a_{i+1}), s(a_i, a_i) \mid 1 \leq i < n\} \cup \{r(a_n, a_1)\}.$$

Each a_i can be identified by $C_i = \exists r^{n-i}. \exists s. \top$, since $a_i \notin C_i^{\mathcal{A}_*^n}$ ($\exists r^k$ is a shorthand for k nestings of the form $\exists r$). E.g., a_1 is the only individual not in $(\exists r. \exists s. \top)^{\mathcal{A}_*^2}$ (see Figure 2).

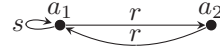


Figure 2: For \mathcal{A}_*^2 , $\mathcal{X} = \{(\mathcal{A}_*, A(a_1)), (\mathcal{A}_*, A(a_2))\}$ is shattered because we can find in \mathcal{L} : $h_1 = \{\exists s. \top \sqcap \exists r. \exists s. \top \sqsubseteq A\}$, $h_2 = \{\exists r. \exists s. \top \sqsubseteq A\}$, $h_3 = \{\exists s. \top \sqsubseteq A\}$, $h_4 = h_2 \cup h_3$.

For all $n \in \mathbb{N}$, $\mathfrak{F}(\mathcal{ELH}, \mathcal{A}_*^n, \text{AQ})$ shatters $\{(\mathcal{A}_*^n, A(a_i)) \mid 1 \leq i \leq n\}$. This does not hold if we add $s(a_n, a_n)$ to \mathcal{A}_*^n . \triangleleft

For discrete cases, in particular, for fragments of first-order logic, the lower bounds obtained with the VC-dimension cannot be larger than the size of the learned expressions assuming a reasonable encoding scheme (Arias and Khardon 2006). The authors argue that many VC-dimension bounds in the literature showing exponential or infinite growth are in terms of some parameters (number of clauses, etc.) determining the size of the target, while other parameters (number of literals, etc.) are ignored. Let $\mathfrak{F}^m = (\mathcal{E}, \mathcal{S}, \mathcal{L}, \mu)$ be a learning framework where the string size of the elements of \mathcal{L} is bounded by m . Since the VC-dimension is bounded by a logarithm of $|\mathcal{L}|$ (for \mathcal{L} finite), $\text{VC}(\mathfrak{F}^m) = O(m)$ (Vapnik 1995).

Proposition 17. *For all $m \in \mathbb{N}$, $\mathfrak{F}^m(\mathcal{ELH}, \mathcal{A}_*, \text{CQ})$ is PAC learnable with a polynomial number of example queries.*

Since the sample complexity (number of classified examples) is polynomial in the size of the target, polynomial time PAC learnability amounts to showing that one can compute a hypothesis in \mathcal{L} that is consistent with the classification of the examples in polynomial time. However, even if \mathcal{A}_* is fixed, checking whether (\mathcal{A}_*, q) is a positive example for a hypothesis \mathcal{H} is NP-hard if the underlying structure of \mathcal{A}_* is non-bipartite (Hell and Nešetřil 1990). So (unless $\text{P} = \text{NP}$) there is not much hope for polynomial time learnability, even with membership queries, since in this case one may not be able to convert the CQ into an IQ (as we did in Theorem 9).

Conclusion

We introduced the OMQA learning setting and investigated the complexity of learning \mathcal{ELH} ontologies in this setting with various query languages. We then considered what happens when the data changes and adaptations to settings where the algorithm learns from classified data, limiting interactions with oracles. Our positive result for IQ-inseparable \mathcal{ELH} TBoxes paves the way for further studies on the complexity of learning ontologies formulated in more expressive languages. We leave the problem of exactly learning \mathcal{ELH} TBoxes with $\text{CQ}_{r,s}$ as an open problem. Learning with a more expressive query language is *not easier* because the oracle can formulate counterexamples which are not informative. Neither it is more difficult because on the other hand, with a more expressive language, the learner can pose more informative membership queries. It would also be interesting to investigate a similar data model in which the ABox is fixed for all the examples, so that the data pool contains examples in the form of queries alone.

Acknowledgments

This research has been supported by the Free University of Bozen-Bolzano through the projects PACO and MLEARN.

References

- Angluin, D. 1988. Queries and concept learning. *Machine Learning* 2(4):319–342.
- Arias, M., and Khardon, R. 2006. Complexity parameters for first order classes. *Machine Learning* 64(1-3):121–144.
- Arias, M.; Khardon, R.; and Maloberti, J. 2007. Learning horn expressions with LOGAN-H. *Machine Learning Research* 8:549–587.
- Arias, M. 2004. *Exact learning of first-order expressions from queries*. Ph.D. Dissertation, Citeseer.
- Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2007a. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition.
- Baader, F.; Ganter, B.; Sertkaya, B.; and Sattler, U. 2007b. Completing description logic knowledge bases using formal concept analysis. In *IJCAI*, 230–235.
- Baader, F.; Lutz, C.; and Brandt, S. 2008. Pushing the EL envelope further. In *OWLED*.
- Bienvenu, M. 2016. Ontology-mediated query answering: Harnessing knowledge to get more from data. In *IJCAI*, 4058–4061.
- Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; and Hellmann, S. 2009. DBpedia – A crystallization point for the Web of Data. *Web Semantics* 7(3):154–165.
- Blum, A. L. 1994. Separating distribution-free and mistake-bound learning models over the boolean domain. *SIAM J. Comput.* 23(5):990–1000.
- Borchmann, D., and Distel, F. 2011. Mining of \mathcal{EL} -GCI. In *ICDM Workshops*, 1083–1090.
- Borchmann, D. 2014. *Learning terminological knowledge with high confidence from erroneous data*. Ph.D. Dissertation, Higher School of Economics.
- Botoeva, E.; Lutz, C.; Ryzhikov, V.; Wolter, F.; and Zakharyashev, M. 2019. Query inseparability for ALC ontologies. *Artif. Intell.* 272:1–51.
- Cimiano, P.; Völker, J.; and Buitelaar, P. 2010. Ontology construction. In *Handbook of Natural Language Processing, Second Edition*. Chapman and Hall/CRC. 577–604.
- Cohen, W. W., and Hirsh, H. 1994. Learning the CLASSIC description logic: Theoretical and experimental results. In *KR*, 121–133.
- Duarte, M. R. C.; Konev, B.; and Ozaki, A. 2018. ExactLearner: A tool for exact learning of \mathcal{EL} ontologies. In *KR*, 409–414.
- Fleischhacker, D.; Völker, J.; and Stuckenschmidt, H. 2012. Mining RDF data for property axioms. In *On the Move to Meaningful Internet Systems: OTM 2012*. Springer. 718–735.
- Frazier, M., and Pitt, L. 1996. Classic learning. *Machine Learning* 25(2-3):151–193.
- Funk, M.; Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2019. Learning description logic concepts: When can positive and negative examples be separated? In *IJCAI*, 1682–1688.
- Ganter, B.; Obiedkov, S. A.; Rudolph, S.; and Stumme, G. 2016. *Conceptual exploration*. Springer.
- Hell, P., and Nešetřil, J. 1990. On the complexity of h-coloring. *J. Comb. Theory Ser. B* 48(1):92–110.
- Klarman, S., and Britz, K. 2015. Ontology learning from interpretations in lightweight description logics. In *ILP*, 76–90.
- Knublauch, H.; Fergerson, R. W.; Noy, N. F.; and Musen, M. A. 2004. The Protégé OWL plugin: An open development environment for semantic web applications. 229–243. Springer.
- Konev, B.; Lutz, C.; Ozaki, A.; and Wolter, F. 2018. Exact Learning of Lightweight Description Logic Ontologies. *Machine Learning Research* 18(201):1–63.
- Konev, B.; Ozaki, A.; and Wolter, F. 2016. A model for learning description logic ontologies based on exact learning. In *AAAI*, 1008–1015.
- Lehmann, J., and Haase, C. 2009. Ideal Downward Refinement in the \mathcal{EL} Description Logic. In *ILP*, 73–87.
- Lehmann, J., and Völker, J. 2014. *Perspectives on Ontology Learning*, volume 18. IOS Press.
- Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic EL. *J. Symb. Comput.* 45(2):194–228.
- Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2012. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press.
- Rudolph, S. 2004. Exploring relational structures via FLE. In *International Conference on Conceptual Structures, ICCS*, 196–212.
- Sazonau, V., and Sattler, U. 2017. Mining hypotheses from data in OWL: advanced evaluation and complete construction. In *ISWC*, 577–593.
- Stuckenschmidt, H.; Parent, C.; and Spaccapietra, S., eds. 2009. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer.
- Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2008. YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics* 6(3):203–217.
- Valiant, L. G. 1984. A theory of the learnable. *Commun. ACM* 27(11):1134–1142.
- Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.
- Völker, J., and Niepert, M. 2011. Statistical schema induction. In *The Semantic Web: Research and Applications*. Springer. 124–138.
- Völker, J.; Fleischhacker, D.; and Stuckenschmidt, H. 2015. Automatic acquisition of class disjointness. *Web Semantics* 35:124–139.
- Vrandečić, D., and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57(10).
- Wong, W.; Liu, W.; and Bennamoun, M. 2012. Ontology Learning from Text: A Look Back and into the Future. *ACM Computing Surveys* 44(4):20:1–20:36.