General Terminology Induction in OWL

Viachaslau Sazonau, Uli Sattler, and Gavin Brown

The University of Manchester
Oxford Road, Manchester, M13 9PL, UK
{sazonauv,sattler,gbrown}@cs.manchester.ac.uk

Abstract. Automated acquisition, or learning, of ontologies has attracted research attention because it can help ontology engineers build ontologies and give domain experts new insights into their data. However, existing approaches to ontology learning are considerably limited, e.g. focus on learning descriptions for given classes, require intense supervision and human involvement, make assumptions about data, do not fully respect background knowledge. We investigate the problem of general terminology induction, i.e. learning sets of general class inclusions, GCIs, from data and background knowledge. We introduce measures that evaluate logical and statistical quality of a set of GCIs. We present methods to compute these measures and an anytime algorithm that induces sets of GCIs. Our experiments show that we can acquire interesting sets of GCIs and provide insights into the structure of the search space.

1 Introduction

An ontology is a machine-processable representation of knowledge about a domain of interest. Ontologies are encoded in formal languages, such as the Web Ontology Language [8], OWL, underpinned by expressive Description Logics, DLs [1]. OWL ontologies are widely-used to represent and share knowledge in application areas such as medicine, biology, astronomy, defence and others.[1] An ontology can contain data and background knowledge (terminology) where both may be incomplete. One might benefit from finding informative correlations in their data taking background knowledge into account. Those correlations may suggest new axioms for the background knowledge or start new inquiries about the data.

However, the problem of terminology induction is generally hard. Firstly, an ideal solution should represent a coherent, self-contained, expert-level modelling. Due to high expressivity of OWL and its Open World Assumption (OWA), the search space can be vast or even infinite depending on the language chosen. Secondly, as usual, the quality of the result depends on the quality of the data which can be incorrect, noisy or insufficient. Ideally, new knowledge should respect the existing knowledge along with the data in order to be maximally informative and avoid contradictions.

Thus, some restrictions and assumptions that simplify the problem are necessary. Another consequence is that any induced knowledge is hypothetical only and requires a domain expert judgement. The contributions of this paper are as follows.

[1] http://bioportal.bioontology.org/

- We state the problem of general terminology induction, i.e. learning sets, called *hypotheses*, of general class inclusions, GCIs, from data (ABox) and background knowledge (TBox).
- We view the problem as multi-objective and define quality criteria for a hypothesis: readability, logical quality, and statistical quality. We define quality measures for a hypothesis that respect the OWA, interactions between axioms in the hypothesis, and interaction of the hypothesis with the background knowledge.
- We have designed and implemented methods to compute the quality measures.
- We have designed, implemented and evaluated an anytime algorithm for general terminology induction. We have gained insights into the structure of the search space and developed heuristics to find out promising hypotheses. The experiments show that we can indeed learn interesting hypotheses.

2 Preliminaries

We assume the reader to be familiar with DLs [1] and OWL [8]. The following nomenclature is used throughout this paper. $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is an ontology where $\mathcal{T}$, $\mathcal{A}$ are TBox and ABox, respectively. N_C, N_R, N_I are disjoint and countably infinite sets of class, property, and individual names, respectively. Σ is a signature, $\widetilde{\mathcal{T}}, \widetilde{\mathcal{A}}, \widetilde{\mathcal{O}}$ are signatures of $\mathcal{T}, \mathcal{A}, \mathcal{O}$, respectively. $ind(\mathcal{O}) = N_I \cap \widetilde{\mathcal{O}}$ is a set of individual names occurring in $\mathcal{O}$. α is a general class inclusion, GCI, also called *axiom*. A, B, X, Y are atomic classes (class names), C, D are complex classes (class expressions), R is a property, a, b, c, d are individuals. $mod(\mathcal{O}, \Sigma)$ is a module [5] of an ontology $\mathcal{O}$ given a signature Σ. $\mathbb{C}$ is a set of (possibly complex) classes. H is a hypothesis, $\mathbb{H}$ is a set of hypotheses. In the following, ABox and TBox are called *data* and *background knowledge*, respectively.

3 Related Work

Ontology learning approaches can be characterised along several dimensions. The first one is a type of the data source, e.g. texts, RDF(S), an oracle (a domain expert), positive and negative examples for a class along with the ABox. The second one is a type of the output knowledge, e.g. class descriptions, class inclusions, and its expressivity. The third dimension is methods used: natural language processing, machine learning, association rules mining, oracle queries, Formal Concept Analysis (FCA), least common subsumer (LCS) computation, etc. The fourth dimension is semantics used that can differ from the OWL semantics, e.g. the Closed World Assumption (CWA). One more characteristic is appreciation of available background knowledge. Finally, the degree of domain expert involvement into the learning process greatly varies across approaches. A survey can be found in [12].

We concentrate on learning from instance-level data, i.e. both class and property assertions. Among the approaches aimed at this type of input data are class description learning, CDL [3, 6, 11], knowledge base completion, KBC [2], association rules mining, ARM [17].

The main method of CDL is machine learning, in particular, Inductive Logic Programming, ILP [13]. The goal is to find a "good" description (class expression) of a

given class name from a set of positive and negative examples [11] for it, i.e. learning is *supervised*. The class description must cover all positive and none of the negative examples. Learning is essentially a search in the space of class expressions guided by refinement operators and heuristics. The background knowledge can be used to optimize the search by exploiting the classification hierarchy. To supervise learning, a domain expert has to provide additional information in form of positive and negative examples for a given class, which can be difficult. As a consequence, there are techniques to sample examples from data. In particular, instances of the class are taken as its positive examples and the CWA is made to obtain its negative examples. However, this way can cause problems [10]. Another method of CDL is finding the least common subsumer (LCS) [3]. LCS is computed from the most specific class (MSC) of each instance of a target class. The method, however, is only applicable to weakly expressive languages.

KBC is based on Formal Concept Analysis (FCA) [7]. It is aimed at acquiring (in some sense) complete knowledge bases, in contrast to CDL. KBC requires to define a set of class expressions in advance which can be hard. The degree of domain expert involvement is high as the expert judges axioms and has to supply a counterexample in the case of rejection. One more limitation is that standard FCA can only be applied under the CWA and the OWA of OWL requires modifications of FCA [2].

ARM is yet another approach to ontology learning [17]. Association rules are mined from transaction tables where columns are predefined class expressions which, similarly to the case of KBC, can be difficult to define in advance. In contrast to KBC, ARM, however, permits acquiring axioms that have counterexamples. In contrast to CDL, ARM induces class inclusions and demands neither positive nor negative examples. The approach focuses on weakly expressive languages. Among other restrictions are its CWA and little appreciation of interaction between induced axioms and the background knowledge, as well as mutual interactions between induced axioms, since they are acquired independently.

Thus, ontology learning approaches simplify the problem in different aspects. As a result, there is no approach that has all following capabilities: learns sets of GCIs, appreciates interactions between axioms within the set and interactions of the set with the background knowledge, uses standard OWL semantics, requires no supervision, does not demand frequent human interventions.

4 Settings and Assumptions

This paper is aimed at addressing the problem of inducing general terminological knowledge from data and background knowledge which together constitute the input ontology. New knowledge is acquired in form of *hypotheses*. A hypothesis is a set of axioms which does not contradict the input ontology, i.e. *consistent* with it, and carries new information, i.e. *informative* for it.

Definition 1. *(Hypothesis) An axiom α is* informative *for an ontology $\mathcal{O}$ if $\mathcal{O} \not\models \alpha$. A set H of axioms (GCIs) is called a* hypothesis *for an ontology $\mathcal{O}$ if H is consistent with $\mathcal{O}$, i.e. $\mathcal{O} \cup H \not\models \top \sqsubseteq \bot$, and each $\alpha \in H$ is informative for $\mathcal{O}$.*

A hypothesis is evaluated by *quality criteria*: *readability*, *statistical quality*, and *logical quality*. Clearly, a hypothesis can be better on one criterion and worse on another. Therefore, we view terminology induction as a multi-objective problem where objectives are *quality measures* corresponding to the quality criteria. Hypotheses are presented to a domain expert who accepts some of them and rejects others. In order to suggest, or *recommend*, good hypotheses first, a preference relation based on quality measures is imposed on the set of hypotheses. In this paper, we apply the following *settings*.

(i) We use OWL and its standard semantics.
  (a) We allow for the usual OWA, i.e. for an instance a and a class C it is possible that $\mathcal{O} \not\models C(a)$ and $\mathcal{O} \not\models (\neg C)(a)$. As a consequence, data can be regarded as just "incomplete".
  (b) Data normally consists of both class and property assertions, e.g. people with family relations, proteins with interactions between them.
  (c) We consider any logic for which subsumption, $\mathcal{O} \models C \sqsubseteq D$, and instance checking, $\mathcal{O} \models C(a)$, are decidable. We use OWL ontologies and reasoners.

(ii) Any input ontology $\mathcal{O}$ is consistent, i.e. data contains no noise which causes inconsistency.

(iii) Learning is *unsupervised*, i.e. no additional information is required in form of positive or negative examples.

(iv) A set $\mathbb{C}$ of target (possibly complex) classes is fixed and finite.

The goal of induction is finding good hypotheses over classes $\mathbb{C}$, or $\mathbb{C}$-*hypotheses*. In the following, we only consider $\mathbb{C}$-hypotheses and omit $\mathbb{C}$ from the name. We also define $\mathbb{C}^- := \mathbb{C} \cup \{\neg C \mid C \in \mathbb{C}\}$.

Definition 2. *($\mathbb{C}$-Hypothesis) Given an ontology $\mathcal{O}$, a hypothesis H for $\mathcal{O}$ is called a* $\mathbb{C}$-hypothesis *if $\alpha \in H$ implies $\alpha = C \sqsubseteq D$, where $C, D \in \mathbb{C}^-$.*

It makes sense to establish a correspondence, sufficient for the task at hand, between an ontology $\mathcal{O}$ and classes $\mathbb{C}$, which we call *projection*.

Definition 3. *(Projection) A* projection π *of an ontology $\mathcal{O}$ to $\mathbb{C}$ is*

$$\pi(\mathcal{O}, \mathbb{C}) := \{D(a) \mid \mathcal{O} \models D(a) \wedge D \in \mathbb{C}^- \wedge a \in ind(\mathcal{O})\}.$$

Thus, a projection is a set of positive and negative class assertions over classes $\mathbb{C}$ entailed by $\mathcal{O}$. A projection can be viewed as a table where rows are labelled with individuals $ind(\mathcal{O})$ and columns are labelled with classes $\mathbb{C}$. Each cell with indices a, C can contain one of three possible values: "1" if $\mathcal{O} \models C(a)$, "0" if $\mathcal{O} \models \neg C(a)$, "?" if $\mathcal{O} \not\models C(a)$ and $\mathcal{O} \not\models \neg C(a)$. Although there are similarities with a transaction table of ARM, our table view is imaginary only and it permits question marks. We will use the table view for better presentation of examples, see Example 1 and Table 1.

Example 1. Given $\mathbb{C} = \{A, B, \exists R.B\}$, $\mathcal{T} = \varnothing$,

$$\mathcal{A} = \{A(a_1), A(a_2), A(a_3), A(a_4), (\neg A)(b), (\neg A)(c), B(c)$$
$$R(a_1, b), R(a_2, b), R(a_3, b), R(a_4, c)\}.$$

We use the projection to evaluate how well a hypothesis fits the *known* data assuming it is correct on the *unknown* data. Indeed, due to the OWA, a hypothesis can make *assumptions* on the unknown data by turning question marks into ones or zeros. If a hypothesis makes too many assumptions, it may be too "strong", e.g. $H = \{\top \sqsubseteq \sqcap_{C \in \mathbb{C}} C\}$. Therefore, it is necessary to evaluate how "brave" a hypothesis is.

	A	B	$\exists R.B$
a_1	1	?	?
a_2	1	?	?
a_3	1	?	?
a_4	1	?	1
b	0	?	?
c	0	1	?

Table 1

Definition 4. *(Assumption) An* assumption *of a hypothesis H in an ontology $\mathcal{O}$ given $\mathbb{C}$ is*

$$\psi(H, \mathcal{O}, \mathbb{C}) := \{D(a) \mid \mathcal{O} \not\models D(a) \wedge \mathcal{O} \cup H \models D(a) \wedge D \in \mathbb{C}^- \wedge a \in ind(\mathcal{O})\}.$$

As a consequence, $\psi(H, \mathcal{O}, \mathbb{C}) \cap \pi(\mathcal{O}, \mathbb{C}) = \varnothing$ for any hypothesis H. Requiring $\mathcal{O} \not\models (\neg D)(a)$ in Definition 4 is not necessary because if $\mathcal{O} \models (\neg D)(a)$ then H is not a hypothesis due its inconsistency with $\mathcal{O}$. Hypotheses making fewer assumptions are preferred according to Occam's razor.

One can think of suggesting hypotheses as single axioms. However, this approach ignores interactions between axioms that can influence the quality of the hypothesis. Two axioms, which are logically "good" individually, do not necessarily create a logically "good" hypothesis. For example, a hypothesis can become *redundant*, e.g. $H = \{A \sqsubseteq B, \neg B \sqsubseteq \neg A\}$, see Section 5.2. In fact, a set of two logically "good" axioms is not necessarily a hypothesis. For example, given that $\{A \sqsubseteq B\}$ and $\{B \sqsubseteq C\}$ are hypotheses for $\mathcal{O}$, a set $\{A \sqsubseteq B, B \sqsubseteq C\}$ is not a hypothesis for $\mathcal{O}$ if $\mathcal{O} \models (A \sqcap \neg C)(a)$. Similar to logical quality, two axioms which are statistically "good" individually may not create a "good" hypothesis which is discussed below, see Section 5.3.

5 Quality Criteria and Measures for a Hypothesis

5.1 Syntactic Length as a Readability Measure

Readability is the ease with which a hypothesis can be read and understood by a human. One of possible measures of readability is the usual *syntactic length* of a hypothesis.

Definition 5. *(Syntactic Length) Let A, C, D be (possibly complex) classes, $A \in N_C$ a class name, $R \in N_R$ a property name, $a \in N_I$ an individual name. The* syntactic length of a GCI *is defined as follows: $|C \sqsubseteq D| := |C| + |D|$, where $|\top| = |\bot| = |A| := 1$, $|\neg C| := 1 + |C|$, $|C \sqcap D| = |C \sqcup D| := 1 + |C| + |D|$, $|\exists R.C| = |\forall R.C| := 1 + |C|$, $|\geqslant nR.C| = |\leqslant nR.C| := 1 + n + |C|$. The* syntactic length of a hypothesis H *is* $|H| := \sum_{\alpha \in H} |\alpha|$.

5.2 Logical Quality

Logical quality evaluates logical properties of a hypothesis: *logical strength* and *redundancy*. Logical strength is commonly called generality in machine learning.

Definition 6. *(Logical Strength) A hypothesis H is* weaker *(more general) than another hypothesis H' if $H' \models H$ and $H \not\models H'$.*

A hypothesis can contain axioms which are superfluous, or *redundant*, within the hypothesis, even if those axioms are informative. For example, axiom $A \sqsubseteq C$ is redundant in hypothesis $\{A \sqsubseteq B, B \sqsubseteq C, A \sqsubseteq C\}$ and axiom $\neg B \sqsubseteq \neg A$ is redundant in hypothesis $\{A \sqsubseteq B, \neg B \sqsubseteq \neg A\}$. Axioms can also have *redundant parts*. For example, D is a redundant part of axiom $A \sqsubseteq C \sqcap D$ in hypothesis $\{A \sqsubseteq B \sqcap D, A \sqsubseteq C \sqcap D\}$.

Definition 7. *(Redundancy) A hypothesis H is* redundant *if there exists a hypothesis H' such that $H' \equiv H$ and $|H'| < |H|$. Otherwise, H is* non-redundant*.*

Lemma 1. *If a hypothesis H is non-redundant, then $|H| = min\{|H'| \mid H' \equiv H\}$.*

We define the logical strength and redundancy of a hypothesis H regardless of $\mathcal{O}$. The reason is that an axiom $\alpha \in H$, which is informative for $\mathcal{O}$ and non-redundant in H, can be interesting, even if it is not informative for $\mathcal{O} \cup H \backslash \{\alpha\}$. Such axiom reveals yet only implicit (and possibly unknown) relation between classes. Additionally, the search for good hypotheses would require entailment checking $\mathcal{O} \cup H \models H'$ which could make it infeasible for hard ontologies.

5.3 Statistical Quality

Statistical quality criteria are aimed at selecting hypotheses that best represent data given background knowledge. In order to comply with the standard OWL semantics and its OWA, we consider the statistical quality of a hypothesis as two-fold. Firstly, hypotheses differently fit data along with background knowledge. Secondly, hypotheses make different number of assumptions in data given background knowledge, i.e. some hypotheses are more cautious than others. Statistically better hypotheses have greater *fitness* and lower *braveness*.

Fitness and Braveness In order to evaluate the statistical quality of a hypothesis, we exploit the idea that axioms can encode regularities in the data. Those regularities can be used to "compress" the data, i.e. to present it in a shorter way. This is the fundamental principle of the *minimum description length induction* [4, 16]. According to it, the better a hypothesis fits the data, the shorter description of the data it provides.

A standard way of measuring the description length is using syntactic measures. However, syntactic measures do not respect logical interactions of a hypothesis with data and background knowledge. Therefore, we introduce a semantic measure of the description length. We define fitness and braveness of a hypothesis as follows.

Definition 8. *(Description Length, Fitness, Braveness) The* description length *of an ABox $\mathcal{B}$ given an ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is*

$$minSize(\mathcal{B}, \mathcal{O}) := min\{|\mathcal{B}'| \mid \mathcal{B}' \cup \mathcal{O} \equiv \mathcal{B} \cup \mathcal{O}\}.$$

Given an ontology $\mathcal{O}$, a set $\mathbb{C}$ of classes, and a hypothesis H, let $\pi := \pi(\mathcal{O}, \mathbb{C})$ and $\psi := \psi(H, \mathcal{O}, \mathbb{C})$. Then

(i) fitness *of* H *is* $fit(H, \mathcal{O}, \mathbb{C}) := |\pi| - minSize(\pi, \mathcal{T} \cup H)$,
(ii) braveness *of* H *is* $bra(H, \mathcal{O}, \mathbb{C}) := minSize(\psi, \mathcal{O})$.

As a consequence of Definition 8, all semantically equivalent hypotheses have the same fitness and the same braveness which is stated by Lemma 2.

Lemma 2. *Given an ontology* $\mathcal{O}$*, a set* $\mathbb{C}$ *of classes, and two hypotheses* H_1, H_2*, if* $H_1 \equiv H_2$ *then* $fit(H_1, \mathcal{O}, \mathbb{C}) = fit(H_2, \mathcal{O}, \mathbb{C})$ *and* $bra(H_1, \mathcal{O}, \mathbb{C}) = bra(H_2, \mathcal{O}, \mathbb{C})$.

Fitness of a hypothesis indicates how well the projection can be shrunk using the hypothesis and background knowledge, i.e. a better shrinkage corresponds to a better fitness. Braveness of a hypothesis measures how many assumptions it makes in the data given the background knowledge. Respecting Occam's razor, hypotheses of lower braveness (or more cautious) are preferred, see Example 2.

	A	B
a	1	1
b	1	?

Table 2

Example 2. The projection π is given by Table 2, $\mathcal{T} = \varnothing$. For $H_1 = \{A \sqsubseteq B\}$ $fit(H_1, \mathcal{O}, \mathbb{C}) = |B(a)| = 1$, $bra(H_1, \mathcal{O}, \mathbb{C}) = |B(b)| = 1$. For $H_2 = \{B \sqsubseteq A\}$ $fit(H_2, \mathcal{O}, \mathbb{C}) = |B(a)| = 1$, $bra(H_2, \mathcal{O}, \mathbb{C}) = 0$. Hence, H_2 is statistically better than H_1.

Two axioms which are statistically "good" individually may or may not create a "good" hypothesis, see Example 3.

	A	B	C
a	?	?	1
b	1	1	1
c	1	1	1

Table 3

Example 3. The projection is given by Table 3, $\mathcal{T} = \varnothing$. Hypotheses $H_1 = \{A \sqsubseteq B\}$, $H_2 = \{B \sqsubseteq C\}$, $H_3 = \{A \sqsubseteq C\}$ are individually statistically confident: $fit(H_1, \mathcal{O}, \mathbb{C}) = fit(H_2, \mathcal{O}, \mathbb{C}) = fit(H_3, \mathcal{O}, \mathbb{C}) = 2$. However, hypothesis $H_{23} = H_2 \cup H_3$ has the same fitness as H_2, H_3: $fit(H_{23}, \mathcal{O}, \mathbb{C}) = 2$. On the other hand, hypothesis $H_{12} = H_1 \cup H_2$ has the fitness twice as big as one of H_1, H_2: $fit(H_{12}, \mathcal{O}, \mathbb{C}) = 4$.

In addition, axioms in the hypothesis can enforce each other, see Example 4.

Example 4. The projection is given by Table 4, $\mathcal{T} = \{B \sqsubseteq C\}$. Hypotheses $H_1 = \{A \sqsubseteq B\}$, $H_2 = \{C \sqsubseteq D\}$ individually have zero fitness. So, the fitness of collective hypothesis $H_{12} = H_1 \cup H_2$ is greater than the total fitness of H_1 and H_2: $fit(H_{12}, \mathcal{O}, \mathbb{C}) = 3$.

Although projection simplifies induction, we may lose some information, in particular, relations between individuals. The latter can result in the overestimation of hypothesis's assumption. In Example 1 let hypothesis $H = \{\neg A \sqsubseteq B\}$, then $\psi(H, \mathcal{O}, \mathbb{C}) = \{B(b), (\exists R.B)(a_1), (\exists R.B)(a_2), (\exists R.B)(a_3)\}$. However, $(\exists R.B)(a_1), (\exists R.B)(a_2), (\exists R.B)(a_3)$ are, in fact, the consequences of $B(b)$ and should not be counted. Braveness correctly handles this: $bra(H, \mathcal{O}, \mathbb{C}) = |\{B(b)\}| = 1$. Illusive assumptions can also be forced by background knowledge and braveness handles this as well, see Example 5.

	A	B	C	D
a	1	?	?	1
b	1	?	?	1
c	1	?	?	1

Table 4

Example 5. The projection π is given by Table 5, $\mathcal{T} = \{B \sqcap C \sqsubseteq D\}$ and $H = \{A \sqsubseteq B, B \sqsubseteq D\}$. The assumption of H is $\psi(H, \mathcal{O}, \mathbb{C}) = \{B(a), B(b), D(a), D(b)\}$ and the braveness is $bra(H, \mathcal{O}, \mathbb{C}) = |\{B(a), B(b)\}| = 2$.

	A	B	C	D
a	1	?	1	?
b	1	?	1	?

Table 5

As a consequence of Definition 8, fitness and braveness are semantically sound and syntax independent measures of the statistical quality of a hypothesis. They take into account both the interaction of a hypothesis with the background knowledge and interactions between axioms within the hypothesis. The measures respect the standard OWL semantics, in particular, they deal with its OWA and, consequently, with incomplete data. Finally, they demand no supervision, such as positive or negative examples, and no additional information besides the input ontology.

Computing Fitness and Braveness Computing fitness and braveness requires finding the size of the minimal projection and assumption, respectively. These may not be unique. All minimal subsets can be found using a hitting set tree algorithm [14]. However, this may require an exponential number of reasoner updates which is computationally expensive given that the fitness and braveness are computed for each hypothesis.

Fortunately, there is a more efficient way to compute the fitness and braveness of a hypothesis avoiding reasoner updates. The idea is to introduce into $\mathcal{O}$ fresh names for classes from $\mathbb{C}^-$, i.e. $\mathcal{O}_X = \mathcal{O} \cup \{X_C \equiv C \mid C \in \mathbb{C}^-\}$, and exploit the inferred class hierarchy of $\mathcal{O}_X$. The function $minSizeUp(\mathcal{B}, \mathcal{O}_X)$ computes an upper bound of the description length $minSize(\mathcal{B}, \mathcal{O})$, which is used for calculating fitness and braveness (see Definition 8):

$$
\begin{aligned}
&minSizeUp(\mathcal{B}, \mathcal{O}_X) := |\mathcal{B}| - |redun(\mathcal{B}, \mathcal{O}_X)|, \text{ where} \\
&\quad redun(\mathcal{B}, \mathcal{O}_X) := \{D(a) \in \mathcal{B} \mid \text{ there is } C \in \tilde{\mathcal{O}}_X \text{ s.t. either} \\
&\qquad \text{(i) } \mathcal{O}_X \models C \sqsubseteq D \wedge \mathcal{O}_X \not\models D \sqsubseteq C \wedge (\mathcal{O}_X \models C(a) \vee C(a) \in \mathcal{B}) \text{ or} \\
&\qquad \text{(ii) } \mathcal{O}_X \models C \equiv D \wedge D \neq unique(D, \mathcal{O}_X)\}, \text{ where} \\
&\quad unique \text{ is a function s.t. } unique(D, \mathcal{O}_X) = D' \text{ implies } \mathcal{O}_X \models D' \equiv D.
\end{aligned}
$$

$minSizeUp(\mathcal{B}, \mathcal{O}_X)$ is based on detecting redundancy of $\mathcal{B}$ given $\mathcal{O}_X$, $redun(\mathcal{B}, \mathcal{O}_X)$, which is the set of those class assertions that can be "easily" inferred from $\mathcal{B} \cup \mathcal{O}_X$ after full classification of $\mathcal{O}_X$. This avoids costly reasoner updates: a reasoner can be executed just once for each hypothesis to classify classes and individuals. However, $minSizeUp(\mathcal{B}, \mathcal{O}_X)$ can overestimate $minSize(\mathcal{B}, \mathcal{O})$ if some redundancy is missed by it. Hence, fitness can be underestimated and braveness can be overestimated, i.e. we may label a hypothesis worse than it is.

6 General Terminology Induction

According to Definition 1, we only consider hypotheses which are logically sound, i.e. informative and consistent with the background knowledge and data. The goal of the induction is finding among those hypotheses ones which have maximal fitness and minimal braveness, or better represent the data.

We impose a readability constraint on a hypothesis: it must not exceed a given syntactic length. The logical weakness of a hypothesis is reflected by its braveness: weaker hypotheses have a lower braveness and are preferred (respecting their fitness)

according to Occam's razor. A redundant hypothesis has the same fitness and braveness as its non-redundant counterpart but a greater length that might be occupied by better axioms. We state the problem of general terminology induction in OWL as follows.

Definition 9. *(General Terminology Induction) Given an ontology $\mathcal{O}$ and a set $\mathbb{C}$ of classes, the problem of* general terminology induction *is to find all best hypotheses which do not exceed length ℓ.*

Thus, as in ILP, we view induction as search in the space of hypotheses restricted by a language bias, determined by $\mathbb{C}$ and ℓ in our case. We regard the process of constructing hypotheses as being equivalent to ranking them in a justified way which is based on fitness and braveness.

6.1 Dominance and Anytime Algorithm

So far, the comparison of hypotheses and terms "better", "best" have not been fully defined. We now define an order on hypotheses via *dominance*.

Definition 10. *(Dominance) Given an ontology $\mathcal{O}$ and a set $\mathbb{C}$ of classes, a hypothesis H* dominates *a hypothesis H', written $H' \prec H$, if $\widetilde{H} = \widetilde{H}'$ and either*

(i) $fit(H,\mathcal{O},\mathbb{C}) > fit(H',\mathcal{O},\mathbb{C}) \wedge bra(H,\mathcal{O},\mathbb{C}) \leqslant bra(H',\mathcal{O},\mathbb{C})$*, or*
(ii) $fit(H,\mathcal{O},\mathbb{C}) \geqslant fit(H',\mathcal{O},\mathbb{C}) \wedge bra(H,\mathcal{O},\mathbb{C}) < bra(H',\mathcal{O},\mathbb{C})$.

By Definition 10 dominance $\prec$ is a strict partial order, i.e. two different hypotheses may be incomparable. *Best* hypotheses are those which are dominated by no other hypotheses. Definition 10 considers only two competitive objectives: fitness and braveness. In addition, we compare hypotheses only if they have the same signature because otherwise interesting hypotheses could be discarded.

The size of the search space depends on $\mathbb{C}$ and ℓ. It varies from $2 \cdot |\mathbb{C}|^2$ (if a hypothesis is restricted to be a single axiom) to $2^{|\mathbb{C}|^2}$ (if a hypothesis is permitted to include all possible axioms). Consequently, the explicit enumeration can be infeasible. We employ an *anytime algorithm*, Algorithm 1, that attempts to explore promising regions of the search space first.

The longer Algorithm 1 runs, the better hypotheses it returns. It can be interrupted at any point which is specified by the termination criteria $stop$, e.g. a timeout, maximal number of iterations, quality threshold, etc. The algorithm processes the whole search space if $stop$ does not prevent it from doing so.

The function $choose(\mathbb{H},\mathcal{O})$ determines which regions of the search space are explored first. Various heuristics can be applied to guide the search. We use the following heuristic for $choose(\mathbb{H},\mathcal{O})$: select $H \in \mathbb{H}$ with maximal

$$q(H,\mathcal{O}) := \frac{1}{|\widetilde{H}|} \cdot \textstyle\sum_{\alpha:=C\sqsubseteq D \,\in\, H} (sup(\alpha,\mathcal{O}) - \rho \cdot [cov(\alpha,\mathcal{O}) - sup(\alpha,\mathcal{O})]),$$

where
$sup(\alpha,\mathcal{O}) := |ins(C \sqcap D,\mathcal{O})|$ is support of α,
$cov(\alpha,\mathcal{O}) := |ins(C,\mathcal{O})|$ is coverage of α,
$ins(C,\mathcal{O}) := \{a \in ind(\mathcal{O}) \mid \mathcal{O} \models C(a)\}$ are instances of C,
$\rho \in (0,\infty)$ is a predefined penalty of "unsupported" coverage.

Algorithm 1 $induceHypotheses(\mathcal{O}, \mathbb{C}, \ell, stop)$

inputs
  $\mathcal{O}$: an ontology
  $\mathbb{C}$: a set of concepts
  ℓ: maximal syntactic length of a hypothesis
  $stop$: termination criteria
outputs
  $\mathbb{H}_{best}$: best hypotheses
do
  $\mathcal{O}_X = \mathcal{O} \cup \{X_C \equiv C \mid C \in \mathbb{C}\}$
  classify $\mathcal{O}_X$ and compute the projection
  $\mathbb{H}_{init} \leftarrow \{\{C \sqsubseteq D\} \mid \{C \sqsubseteq D\}$ is a hypothesis $\wedge\ C, D \in \mathbb{C}^-\}$
  $\mathbb{H} \leftarrow \mathbb{H}_{init}$, $\mathbb{H}_{best} \leftarrow \varnothing$
  while $\mathbb{H} \neq \varnothing$ and $stop$ is not satisfied **do**
    $H \leftarrow choose(\mathbb{H}, \mathcal{O}_X)$
    $\mathbb{H} \leftarrow \mathbb{H} \backslash \{H\}$
    classify $\mathcal{O}_X \cup H$ and compute the assumption of H
    compute fitness and braveness of H using $minSizeUp$
    $\mathbb{H}_{best} \leftarrow \mathbb{H}_{best} \cup \{H\}$
    if H is not complete **then** % extensions are possible
      $\mathbb{H}_{ext} \leftarrow \{H \cup H' \mid H' \in \mathbb{H}_{init} \wedge |H \cup H'| \leqslant \ell \wedge H \cup H' \notin \mathbb{H} \cup \mathbb{H}_{best}\}$
      $\mathbb{H} \leftarrow \mathbb{H} \cup \mathbb{H}_{ext}$ % add all direct extensions of H
    end if
  end while
  remove dominated hypotheses from $\mathbb{H}_{best}$
  return $\mathbb{H}_{best}$

The heuristic chooses hypotheses that have smaller signatures and consist of axioms with larger support and smaller unsupported coverage. More importantly, it forces Algorithm 1 to firstly explore hypotheses with connected axioms (due to $1/|\widetilde{H}|$) of higher independent statistical quality. The higher the penalty ρ is, the more likely it is for cautious hypotheses to be evaluated first. If Algorithm 1 enumerates the full search space, then the heuristic does not affect the outcome. Only in this case Algorithm 1 is guaranteed to be complete.

Although a reasoner is updated just once per hypothesis, computing the fitness and braveness can still be expensive if the ontology is computationally hard. This can result in a small number of evaluated hypotheses once the termination criteria $stop$ are satisfied. Incremental reasoners, such as FaCT++ [15], can improve the performance if a hypothesis is not big. Hence, besides readability and size of the search space, the length of a hypothesis may affect the performance of computing its fitness and braveness.

6.2 Choice of Classes

So far, we have assumed that a set $\mathbb{C}$ of interesting classes is known. For example, it can be defined by a domain expert. Unfortunately, this can be a difficult problem on its own. There are several possibilities to automate the choice of target classes. First, one can extract all subclasses, including complex ones, occurring in the ontology $\mathcal{O}$.

These are suitable candidates because they are explicitly asserted in the ontology which implies that a domain expert is more likely to find them sensible and interesting.

However, an ontology can have poor terminological knowledge, in particular, it can contain mostly atomic classes. In this case, classes $\mathbb{C}$ can be generated from some signature $\Sigma \subseteq \widetilde{\mathcal{O}}$ using a target *class language*, see Example 6.

Example 6. The signature is $\Sigma = \{A_1, A_2, R_1, R_2\}$ and target class language is $G = \{X \mid X \in \Sigma\} \cup \{X \sqcap Y \mid X, Y \in \Sigma\} \cup \{\exists R.X \mid X \in \Sigma\}$ (OWL's structural equivalence is employed to avoid duplicates). Then, the set of classes is generated as follows: $\mathbb{C} := \{A_1, A_2, A_1 \sqcap A_2, \exists R_1.A_1, \exists R_1.A_2, \exists R_2.A_1, \exists R_2.A_2\}$.

If the ontology signature is large and our class language is expressive, the produced set of class expressions can be vast. One way to deal with the problem is to determine unpromising classes in $\mathbb{C}$ and discard them. Another way is to select a signature of interest $\Sigma \subset \mathcal{O}$ of manageable size and construct classes $\mathbb{C}$ from it using a language G. Σ can be specified by a domain expert which may be hard due to the lack of knowledge, large ontology signature, etc. Alternatively, Σ can be selected automatically.

Since we run our experiments on OWL ontologies which we are not familiar with and do not have access to domain experts, we select a signature Σ of an ontology $\mathcal{O}$ with respect to $\mathcal{A}$ using the modular structure of the ontology as follows: $\Sigma := \widetilde{M}$, where $M = module(\mathcal{T}, \widetilde{\mathcal{A}})$ (we use $\top\bot$-modules [5]).

This approach yields class and property names that are logically connected with $\mathcal{A}$ and discards logically disconnected ones (those can be numerous). We construct classes $\mathbb{C}$ from Σ using a language G. Finally, we discard classes from $\mathbb{C}$ that have no instances.

7 Implementation and Evaluation

7.1 Implementation

Tools and Hardware All algorithms are implemented in Java 7 using OWL API (3.5.0). We use the OWL 2 DL reasoner FaCT++ (1.6.3) [15] which supports incremental reasoning. The experiments are executed on the following machine: Linux Ubuntu 14.04.2 LTS (64 bit), Intel Core i5-3470 3.20 GHz, 8 GB RAM.

7.2 Evaluation

Evaluation Goals By Definition 9, the solution of the general terminology induction problem is a set of hypotheses. It depends on the following parameters: an ontology $\mathcal{O}$, a set $\mathbb{C}$ of classes, and a maximal length ℓ. The evaluation aim is to empirically assess the influence of these parameters on the solution. More specifically, the experiments are aimed at answering the following questions.

- Q1 Where are we likely to find good hypotheses: in more expressive languages for $\mathbb{C}$ or bigger values of ℓ?
- Q2 How does expressivity of the language and maximal length of a hypothesis influence the performance of computing the fitness and braveness?
- Q3 Can we acquire hypotheses that seem plausible, so that we can use them to enrich our background knowledge, or that tell us interesting information about our data?

Choice of Ontologies We conduct the empirical evaluation on a corpus of ontologies selected from related work [10, 6] including DL-Learner datasets,[2] Protégé OWL,[3] and TONES[4] repositories. The Kinship ontology is obtained from UCI Machine Learning Repository.[5] We have selected the ontologies based on the following criteria. Firstly, data contains both class and property assertions, at least 15 individuals. Secondly, ontology classification takes less 10 minutes. Thirdly, we are sufficiently confident that we understand the topic of the ontology. The corpus is available online.[6]

Table 6 describes the corpus where we use the following metrics. $|ind(\mathcal{A})|$, CA, RA are numbers of individuals, concept and property assertions in the ABox, respectively. $degree(\mathcal{A})$, $conn(\mathcal{A})$ are the average degree and average number of individuals in a connected component, respectively. $|\widetilde{\mathcal{A}}|$, $|\widetilde{\mathcal{T}}|$ are sizes of the ABox and TBox signature. $Jac(\widetilde{\mathcal{A}}, \widetilde{\mathcal{T}})$ is the Jaccard index of ABox and TBox signatures, $open(\mathcal{A}, \mathcal{T})$ is the average number of question marks per individual-class name pair.

	DL	$\lvert ind(\mathcal{A})\rvert$	CA	RA	$degree(\mathcal{A})$	$conn(\mathcal{A})$	$\lvert\widetilde{\mathcal{A}}\rvert$	$\lvert\widetilde{\mathcal{T}}\rvert$	$Jac(\widetilde{\mathcal{A}}, \widetilde{\mathcal{T}})$	$open(\mathcal{A}, \mathcal{T})$
Alzheimer	$\mathcal{AL}$	150	106	854	5.7	150	40	0	0	0.96
Arch	$\mathcal{ALC}$	19	26	26	1.4	3.8	10	13	0.77	0.53
BasicFamily	$\mathcal{ALI}$	31	50	95	3.1	10.3	6	6	1	0.67
Carcinogenesis	$\mathcal{ALC}(\mathcal{D})$	22372	22372	40666	1.8	65.8	113	146	0.77	0.65
Cinema	$\mathcal{ALCOF}$	45	45	76	1.7	45	7	37	0.19	0.88
Earthrealm	$\mathcal{SHOIN}(\mathcal{D})$	171	179	203	1.2	7.4	23	2482	0.01	0.89
Economy	$\mathcal{ALCH}(\mathcal{D})$	482	649	555	1.2	5.3	29	380	0.04	0.94
Financial	$\mathcal{ALCOIF}$	17941	17941	47248	2.6	8970.5	52	76	0.68	0.54
GeoSkills	$\mathcal{ALCHOIN}(\mathcal{D})$	2592	4681	3896	1.5	13.9	569	618	0.90	0.69
Heart	$\mathcal{AL}(\mathcal{D})$	280	275	1080	3.9	280	9	11	0.82	0.90
Kinship	$\mathcal{ALI}$	24	116	40	1.7	12	18	4	0.16	0.81
KRK	$\mathcal{SHI}$	420	525	1508	3.6	4	25	40	0.55	0.65
Mammographic	$\mathcal{AL}(\mathcal{D})$	975	975	2883	3.0	975	18	22	0.82	0.97
MDM073	$\mathcal{ALCHOF}(\mathcal{D})$	112	130	169	1.5	2.0	82	215	0.38	0.51
Mutagenesis	$\mathcal{AL}(\mathcal{D})$	14145	14145	26533	1.9	61.5	60	91	0.66	0.99
NTN	$\mathcal{SHOIN}(\mathcal{D})$	724	724	1636	2.3	2.8	64	78	0.82	0.96
Suramin	$\mathcal{AL}(\mathcal{D})$	2979	2979	6008	2.0	175.2	20	49	0.41	0.97

Table 6: Ontologies and their metrics

Evaluation Setup To answer the raised questions, we set up the following experimental pipeline. Given an ontology $\mathcal{O}$, for each combination of a class language G and maximal length ℓ we run Algorithm 1 with the timeout $stop$ set to 10 minutes. Once Algorithm 1

[2] https://github.com/AKSW/DL-Learner

[3] http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library

[4] http://owl.cs.manchester.ac.uk/repository/

[5] https://archive.ics.uci.edu/ml/datasets/Kinship

[6] http://www.cs.man.ac.uk/~sazonauv/tbox_induction/corpus/

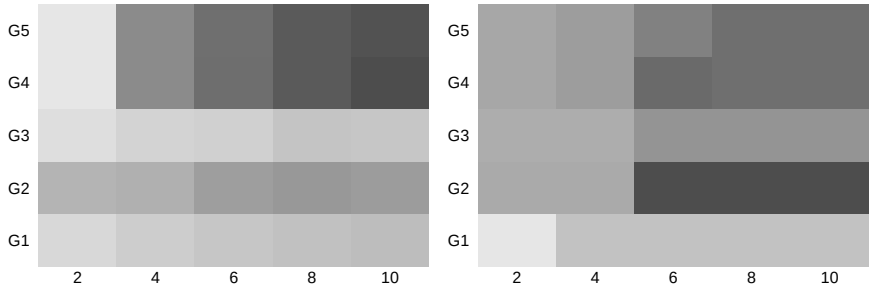
terminates, we record the fitness and braveness of each hypothesis in the output set. We also record the average hypothesis evaluation time which comprises computing the fitness and braveness. Finally, we store all hypotheses if their number is less than 100 and only 100 hypotheses of maximal $q(H, \mathcal{O})$ otherwise.

We choose maximal length ℓ from $\{2, 4, 6, 8, 10\}$. In order to generate classes $\mathbb{C}$, we use the process described in Section 6.2. The signature is $\Sigma := \widetilde{M}$, where $M = module(\mathcal{T}, \widetilde{\mathcal{A}})$. We investigate 5 class languages G_i, such that $G_i \subseteq G_{i+1}$ (duplicates are avoided by the means of OWL's structural equivalence):

$G_1 := \{X \mid X \in \Sigma\}$;
$G_2 := G_1 \cup \{X_M \mid X_M$ is a possibly complex subclass in $M\}$;
$G_3 := G_2 \cup \{X \sqcap Y \mid X, Y \in \Sigma\}$;
$G_4 := G_3 \cup \{\exists R.X \mid X, R \in \Sigma\}$;
$G_5 := G_4 \cup \{X \sqcap \exists R.Y \mid X, Y, R \in \Sigma\}$.

7.3 Results

Dependence of fitness and braveness on language and length is shown on Figure 1. For each ontology the experiment is executed as described above. The values obtained are normalised, i.e. divided by the maximal value. Then, the values are aggregated across the corpus and the average value is reported per cell.



(a) Fitness (b) Braveness

Fig. 1: Dependence of fitness (a) and braveness (b) on language expressivity and maximal length: darker colours reflect greater numbers

Our first observation is that some languages and lengths result in no hypotheses induced which happens if a class language is not expressive enough or hypothesis length is too low. We aggregate and average only over non-empty values. An expected observation is that increasing expressivity is useless if an ontology is poor, e.g. contains few relations in the data and axioms in the background knowledge. On the other hand, if an ontology is rich, increasing expressivity may or may not be fruitful.

Figure 1 shows that increasing length always results in hypotheses of higher fitness and mostly, but not always, of higher braveness since added axioms may make no assumptions or repeat the assumptions already made. Increasing expressivity also generally leads to higher fitness and higher braveness. However, the changes are not as gradual as for length, in particular, braveness seems irregular. Interestingly, we observe

that G_2 consistently outperforms G_3 in fitness, despite $G_2 \subseteq G_3$, which can be explained as follows. On the one hand, the search space considerably increases from G_2 to G_3. On the other hand, G_3 appears to be less fruitful than G_2 (compare to G_4 and G_5). As a result, it becomes harder to find equally good hypotheses in the same time. Thus, the answer to Q2 is that increasing expressivity and length promises better fitness but commonly worse braveness.

We also observe that the average hypothesis evaluation time does not vary widely. Thus, the answer to Q2 is that performance does not degrade significantly for higher expressivity and length. The performance of evaluating a hypothesis is as follows: less than 0.1 second for 8 ontologies, from 0.1 to 1 second for 4 ontologies, from 1 to 10 seconds for 4 ontologies, and around 15 seconds for 1 ontology. The results can be found online.[7]

Ontology	Examples of hypotheses
Alzheimer	$Drug \sqsubseteq \exists getsReplacedBy.Substituent$ $Substituent \sqsubseteq \exists hasPolatisation.Polar$ $\exists hasPolatisation.Polar \sqsubseteq \exists isHAcceptor.HAcceptor$
Arch	$construction \sqsubseteq \exists hasPillar.pillar$ $\exists hasParallelpipe.wedge \sqsubseteq \exists hasPillar.freeStandingPillar$ $\exists touches.pillar \sqsubseteq \exists leftof.pillar$
BasicFamily	$\exists hasChild.Person \sqsubseteq Person$ $\exists hasParent.Person \sqsubseteq Person$ $\exists hasParent.Female \sqsubseteq \exists hasParent.Male$
Cinema	$Movie \sqsubseteq \exists hasForActor.Actor$ $Movie \sqsubseteq \exists hasForGenre.Genre$ $\exists hasForActor.\{Eastwood\} \sqsubseteq \exists hasForGenre.\{Western\}$ $\exists hasForDirector.\{Burton\} \sqsubseteq \exists hasForActor.\{Depp\}$
Earthrealm	$\exists hasDefaultUnit.BaseUnit \sqsubseteq \exists hasDefaultUnit.ComplexUnit$ $\exists hasDefaultUnit.\{second\} \sqsubseteq TimeRelatedQuantity$ $\exists hasDefaultUnit.\{meterPerSecond\} \sqsubseteq DrySeasonDuration$
Economy	$Nation \equiv IndependentState$ $\exists economyType.EconomicDevelopmentLevel$ $\sqsubseteq \exists economyType.IMFDevelopmentLevel$
Financial	$Account \sqsubseteq \exists hasStatementIssuanceFrequency.Monthly$ $\exists isOwnerOf.Account \sqsubseteq Client$
Mammographic	$\exists hasMargin.spiculated \sqsubseteq \exists hasShape.irregular$ $\exists hasShape.irregular \sqsubseteq \exists hasDensity.low$
Mutagenesis	$Compound \sqsubseteq \exists hasBond.Bond1$ $\exists inBond.Hydrogen3 \sqsubseteq Bond1$ $\exists inBond.Oxygen40 \sqsubseteq \exists inBond.Nitrogen38$
NTN	$Man \equiv \forall spouseOf.Woman$ $\exists knows.Man \sqsubseteq Man$ $\exists relativeOf.Man \sqsubseteq Man$

Table 7: Examples of hypotheses induced within 10 minutes

In order to answer Q3, we act as domain experts and eyeball the induced hypotheses. We aim at finding plausible and interesting hypotheses. Some results are shown in Table 7. Firstly, we observe that induced hypotheses can, in fact, enrich the background knowledge, see Table 7. If the background knowledge is poor, as in BasicFamily and

[7] http://www.cs.man.ac.uk/~sazonauv/tbox_induction/results/

Cinema, or even absent, as in Alzheimer, hypotheses seem to be a good starting point for modellers. If the background knowledge is incomplete, hypotheses appear to be interesting missing bits, e.g. for Economy, Financial, NTN, and Mutagenesis.

Secondly, we observe that hypotheses can reveal interesting relations in our data. This can expose new knowledge about the domain and help to understand the data. For example, hypotheses discover relations between particular actors, directors, and movie genres from Cinema. Another example is Mammographic where we can learn relations between diagnostic observations, e.g. having irregular shape implies having lower density. Such hypotheses can potentially inform doctors of yet unknown relations in their data, facilitate future research in the domain, and lead to data improvements, e.g. a supplement of images of tumours that have irregular shape and high density.

Thirdly, hypotheses can contain "strange" axioms which may help us highlight, on the one hand, odd or erroneous modelling and, on the other hand, inaccurate or abnormal data. We observe this for Arch inducing $\exists touches.pillar \sqsubseteq \exists leftof.pillar$ (why is there nothing to the right?) and for Earthrealm inducing $\exists hasDefaultUnit.\{meterPerSecond\} \sqsubseteq DrySeasonDuration$ (wrong unit?). Thus, we can answer Q3 positively.

Although we use different settings and the goal of induction is different, we make some comparison of our results with related work. In particular, we consider the supervised CDL and its implementation DL-Learner [11]. Given a set of positive and negative examples for a target class $construction$ in Arch, it searches for definition $construction \equiv \exists hasPillar.(freeStandingPillar \sqcap \exists leftof.\exists supports.\top)$. As Table 7 shows, our approach induces a weaker definition of $construction$ along with some related knowledge. For Cinema we observe that descriptions of different movie types are induced, e.g. $EastwoodMovie \sqsubseteq \exists hasForActor.\{Eastwood\}$, $EastwoodMovie \sqsubseteq \exists hasForGenre.\{Western\}$. For NTN the definition $Man \equiv \forall spouseOf.Woman$ is induced. Thus, although our approach is unsupervised, it shows the potential to learn class definitions.

8 Discussion and Future Work

The evaluation shows that our approach is able to induce interesting hypotheses. On the one hand, they can potentially be helpful to build and improve the background knowledge. On the other hand, hypotheses seemingly discover new knowledge about the domain and help us understand the data. Interestingly, they may help us identify modelling errors and data flaws.

Although the search space is vast, general terminology induction is feasible. It is encouraging given that statistically and logically sound measures are used to evaluate a hypothesis and this requires reasoning. We observe that larger and more expressive hypotheses are generally better and still feasible.

As for future work, we will investigate more informed ways of constructing a set of promising initial classes, e.g. using techniques from CDL, along with new algorithms and heuristics for search space exploration. We will also attempt to extend the methodology to deal with noisy data that causes inconsistency, e.g. using techniques from [9]. We plan to investigate learning property hierarchies.

We intend to go beyond the corpus and carry out case studies with domain experts to evaluate our approach in more detail. We also consider other scenarios, e.g. how acceptance or rejection of a hypothesis affects other hypotheses, how hypotheses can be used for predicting class memberships of individuals, terminology abduction and "what if" analysis of data under the OWA.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing Description Logic knowledge bases using formal concept analysis. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. pp. 230–235. IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007)
3. 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)
4. Conklin, D., Witten, I.H.: Complexity-based induction. Machine Learning 16(3), 203–225 (1994)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research pp. 273–318 (2008)
6. Fanizzi, N., D'Amato, C., Esposito, F.: DL-FOIL concept learning in Description Logics. In: Proceedings of the 18th International Conference on Inductive Logic Programming. pp. 107–121. ILP'08, Springer-Verlag, Berlin, Heidelberg (2008)
7. Ganter, B., Wille, R.: Formal Concept Analysis, vol. 284. Springer Berlin (1999)
8. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Web Semantics 6(4), 309–322 (2008)
9. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Proceedings of the Second European Conference on The Semantic Web: Research and Applications. pp. 182–197. ESWC'05, Springer-Verlag, Berlin, Heidelberg (2005)
10. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. Web Semantics 9(1), 71–81 (2011)
11. Lehmann, J., Hitzler, P.: Concept learning in Description Logics using refinement operators. Machine Learning 78(1-2), 203–250 (2010)
12. Lehmann, J., Völker, J. (eds.): Perspectives On Ontology Learning, Studies in the Semantic Web, vol. 18. IOS Press (2014)
13. Muggleton, S.: Inductive logic programming. New Generation Computing 8(4), 295–318 (1991)
14. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
15. Tsarkov, D., Horrocks, I.: FACT++ Description Logic reasoner: System description. In: Proceedings of the 3rd International Joint Conference on Automated Reasoning. Lecture Notes in Computer Science, vol. 4130, pp. 292–297. Springer-Verlag (2006)
16. Vitányi, P.M., Li, M.: Minimum description length induction, bayesianism, and kolmogorov complexity. IEEE Transactions on Information Theory 46(2), 446–464 (2000)
17. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) The Semantic Web: Research and Applications, Lecture Notes in Computer Science, vol. 6643, pp. 124–138. Springer Berlin Heidelberg (2011)