Closed-World Concept Induction for Learning in OWL Knowledge Bases

David Ratcliffe^{1,2} and Kerry Taylor^{2,1}

¹ College of Engineering and Computer Science
Australian National University, Canberra ACT 2601, Australia

² CSIRO Digital Productivity
CS & IT Building 108, North Road, Australian National University
GPO Box 664, Canberra, ACT 2601, Australia
{david.ratcliffe,kerry.taylor}@csiro.au

Abstract. We present a general-purpose method for inducing OWL class descriptions over data and knowledge captured with RDF and OWL in a closed-world way. We combine our approach with a top-down refinement-based search with Description Logic (DL) expressions which incorporates OWL background knowledge. Our methods are designed for speed and scalability to support analysis tasks like data mining over large knowledge-rich data sets. We compare our methods to a state-of-the-art DL learning tool with respect to a large benchmark problem to demonstrate the speed and effectiveness of our approach.

Keywords: concept induction, data mining, machine learning, DL, RDF, OWL.

1 Introduction

1.1 Motivation

Semantic Web technologies such as the Resource Description Framework (RDF) data model and the Web Ontology Language (OWL) have experienced wide uptake in recent years, particularly within the life sciences. Of particular interest to us are domains which seek to collect and organise large amounts of data and knowledge using RDF and OWL to aid experts in exploring their data. For example, the Crystallisation Data Exchange (XDX) Consortium in the field of protein crystallisation in structural biology aims to collect and integrate experimental data and knowledge with an OWL ontology to help domain experts construct hypotheses about experimental conditions which might be efficient crystallisation catalysts [1]. Similarly, the Kidney and Urinary Pathway Database (KUPKB) collects and organises data and knowledge with an OWL ontology relating to experiments in renal biology to help domain experts construct hypotheses about the causes of renal complications [2]. In domains such as these, automating hypothesis generation in light of data and formal background knowledge captured with OWL is possible with data mining and machine learning methods which have the potential to aid in the search for new scientific knowledge.

Methods and systems have emerged in recent years for automatically inducing OWL concept expressions from RDF instance data. The goal of such methods so far has been to produce OWL concept descriptions which perfectly characterise a class for integration into an ontology, or which induce concepts to capture a classification rule or an interesting cluster in RDF data. These methods have the benefit of utilising expressive background ontologies to structure the search space and to provide an expressive yet comprehensible domain-specific vocabulary with which to pose hypotheses. Many of these such methods bring to bear theoretical results from concept learning in Inductive Logic Programming to Description Logics which underpin the formal semantics of OWL. Most notably, top-down refinement-based search methods have been explored with the DL-LEARNER [3], YINYANG [4], DL-FOIL [5] and FR-ONT [6] systems. Besides DL-LEARNER, these systems use hypothesis languages with limited expressivity and base their learning methods on instance checking under the open-world assumption which can be computationally expensive.

In our work, we aim to improve upon the expressivity, speed and scalability of these methods so as to make them viable for data mining large RDF/OWL knowledge-bases. To achieve this, we have focussed on constraining the search space of OWL expressions under a *closed-world assumption* (CWA), which is commonly employed in machine learning and data mining, yet is uncommon for use with OWL. In this paper, we formally characterise a closed-world interpretation for concept induction in the expressive concept language known as *ALCOQ* and describe how it can be used to limit the search space to improve speed and scalability of the search for concept expressions. We also describe how induction with a closed-world assumption can lead to concepts which are unsatisfiable under an open-world assumption with respect to a background ontology.

2 Concept Learning in Description Logics

The setting in which we are considering concept induction involves an OWL background ontology with a set of RDF data which are assertions of instances of various concepts and roles in the ontology. Amongst these is a set of classes $\mathbf{P} = \{P_1, \dots, P_n\}$ for $n \geq 1$ which are called *label classes* which capture disjoint sets of named instances (RDF resources) which are *examples* for a learning problem. In the *supervised machine learning classification* ($n > 1$) setting, we usually aim to induce concept expressions which describe examples from one label class and none other. *Unsupervised learning* ($n = 1$) aims to induce a hypothesis which captures a significant proportion of a set of examples (e.g. a *cluster*).

The remainder of this section is organised as follows. §2.1 formally introduces Description Logics (DLs); §2.2 describes a closed-world interpretation for use in learning expressive DL expressions; §2.3 introduces a refinement-based search method for concept induction, and §2.4 concludes with some considerations for closed-world concept induction against open-world background knowledge.

2.1 Description Logics

Description Logics (DL) are a family of knowledge representation languages based on decidable fragments of first-order logic [7]. The formal semantics of OWL is underpinned by various DLs, such as *SRQIQ* which provides semantics for the OWL2-DL profile¹. OWL classes and properties are described as concepts and roles (respectively) in DL terminology, which we will now use interchangeably throughout this paper. A DL *knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{A}$ is called the *assertional box* (ABox) which contains a set of *assertions* declaring named individuals to be known as *instances* of *concepts* such as $Car(c_0)$, $Engine(e_0)$ and to associate individuals with roles, such as $hasEngine(c_0, e_0)$. $\mathcal{T}$ is called the *terminological box* (TBox) containing a set of *inclusion axioms* which assert implications between instances belonging to one concept (role) as also belonging to another concept (role) in the form $C \sqsubseteq D$, denoting that all instances (tuples) of concept (role) C imply membership in concept (role) D , for example $Car \sqsubseteq Vehicle$ ($hasEngine \sqsubseteq hasPart$). An *interpretation* is a pair $(\Delta^{\mathcal{I}}, \mathcal{I})$ where $\Delta^{\mathcal{I}}$ represents the set of all domain individuals and $\mathcal{I}$ is a mapping from any concept C to a subset of the domain $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, any role R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and individuals i to an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. For example, the instances $c_0^{\mathcal{I}}, e_0^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, $c_0^{\mathcal{I}} \in Car^{\mathcal{I}}$, $e_0^{\mathcal{I}} \in Engine^{\mathcal{I}}$ and the tuple $\langle c_0^{\mathcal{I}}, e_0^{\mathcal{I}} \rangle \in hasEngine^{\mathcal{I}}$. We say that an interpretation $\mathcal{I}$ is a *model* of (or, *satisfies*) $\mathcal{T}$ if, for all axioms $C \sqsubseteq D$ in $\mathcal{T}$, we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. A knowledge base $\mathcal{K}$ is said to *entail* some assertion $C(i)$ as $\mathcal{K} \models C(i)$ if, for all interpretations $\mathcal{I}$, $i \in C^{\mathcal{I}}$ (also, $\mathcal{I} \models C(i)$).

2.2 Closed-World DL Interpretation

An *open-world assumption* (OWA) is typically made of the DLs which underpin OWL knowledge-bases, in that if $\mathcal{K} \not\models \psi$, then it does not follow that $\mathcal{K} \models \neg\psi$ which would otherwise follow under a *closed-world assumption* (CWA). For example, if it cannot be shown that $x^{\mathcal{I}} \in C^{\mathcal{I}}$, then we do not assume that $x^{\mathcal{I}} \in (\neg C)^{\mathcal{I}}$, yet under a closed-world assumption the latter would be implied.

An appropriate closed-world interpretation which is applicable to the expressive DL known as *SRQIQ* (corresponding to OWL2-DL) knowledge bases has been described by Tao et. al. [8] as the *IC-interpretation* for handling integrity constraints in OWL. This interpretation is defined as the pair $(\mathcal{I}, \mathcal{U})$ where $\mathcal{I}$ is a *SRQIQ* interpretation defined over the domain $\Delta^{\mathcal{I}}$ and $\mathcal{U}$ is the set of all first-order models of a *SRQIQ* knowledge base. The interpretation function $\cdot^{\mathcal{I}, \mathcal{U}}$ maps concepts and roles to sets as follows, where the set N_I contains the named individuals of $\mathcal{K}$:

$$\begin{aligned} C^{\mathcal{I}, \mathcal{U}} &= \{x^{\mathcal{I}} : x \in N_I \text{ s.t. } \forall \mathcal{J} \in \mathcal{U}, x^{\mathcal{J}} \in C^{\mathcal{J}}\} \\ R^{\mathcal{I}, \mathcal{U}} &= \{\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle : x, y \in N_I \text{ s.t. } \forall \mathcal{J} \in \mathcal{U}, \langle x^{\mathcal{J}}, y^{\mathcal{J}} \rangle \in R^{\mathcal{J}}\} \end{aligned}$$

The interpretations $C^{\mathcal{I}, \mathcal{U}}$ and $R^{\mathcal{I}, \mathcal{U}}$ contain only what is *known* in $\mathcal{K}$, and all other known individuals (or tuples thereof) are assumed to lie in the negation $(\neg C)^{\mathcal{I}, \mathcal{U}}$

¹ See <http://www.w3.org/TR/owl2-direct-semantics/> for more details.

and $(\neg R)^{\mathcal{I}, \mathcal{U}}$ respectively. The interpretation extends to arbitrary concept expressions in $\mathcal{ALCCOQ}$ inductively as follows, where $\#$ denotes set cardinality:

$$\begin{aligned}
\top^{\mathcal{I}, \mathcal{U}} &= \Delta^{\mathcal{I}, \mathcal{U}} \\
\perp^{\mathcal{I}, \mathcal{U}} &= \emptyset \\
(\neg A)^{\mathcal{I}, \mathcal{U}} &= \Delta^{\mathcal{I}, \mathcal{U}} \setminus A^{\mathcal{I}, \mathcal{U}} \\
(C \sqcap D)^{\mathcal{I}, \mathcal{U}} &= C^{\mathcal{I}, \mathcal{U}} \cap D^{\mathcal{I}, \mathcal{U}} \\
(C \sqcup D)^{\mathcal{I}, \mathcal{U}} &= C^{\mathcal{I}, \mathcal{U}} \cup D^{\mathcal{I}, \mathcal{U}} \\
\{x\}^{\mathcal{I}, \mathcal{U}} &= \{x : x \in \Delta^{\mathcal{I}, \mathcal{U}}\} \\
(\exists R.C)^{\mathcal{I}, \mathcal{U}} &= \{x : x \in \Delta^{\mathcal{I}, \mathcal{U}} \text{ s.t. } \exists y. \langle x, y \rangle \in R^{\mathcal{I}, \mathcal{U}} \wedge y \in C^{\mathcal{I}, \mathcal{U}}\} \\
(\forall R.C)^{\mathcal{I}, \mathcal{U}} &= \{x : x \in \Delta^{\mathcal{I}, \mathcal{U}} \text{ s.t. } \forall y. \langle x, y \rangle \in R^{\mathcal{I}, \mathcal{U}} \rightarrow y \in C^{\mathcal{I}, \mathcal{U}}\} \\
(\geq^n R.C)^{\mathcal{I}, \mathcal{U}} &= \{x : x \in \Delta^{\mathcal{I}, \mathcal{U}} \text{ s.t. } \#\{y. \langle x, y \rangle \in R^{\mathcal{I}, \mathcal{U}} \wedge y \in C^{\mathcal{I}, \mathcal{U}}\} \geq n\} \\
(\leq^n R.C)^{\mathcal{I}, \mathcal{U}} &= \{x : x \in \Delta^{\mathcal{I}, \mathcal{U}} \text{ s.t. } \#\{y. \langle x, y \rangle \in R^{\mathcal{I}, \mathcal{U}} \wedge y \in C^{\mathcal{I}, \mathcal{U}}\} \leq n\}
\end{aligned}$$

This interpretation has several properties which make it suitable for efficient concept induction for data mining and machine learning in DL knowledge-bases. Firstly, the closed-world assumption is often used in data mining and machine learning, as we commonly assume that knowledge of our example instance data is complete. Secondly, concept learning algorithms which generate hypotheses typically evaluate them in terms of their *coverage* over the example data. In a DL setting, hypotheses as concept expressions are checked for their example coverage by *instance checking*. With an open-world assumption, instance checking is performed by entailment which can be computationally expensive for knowledge-bases with expressive background ontologies, such as N2ExpTime-complete for OWL2-DL based on $\mathcal{SROIQ}$ [9]. Learning algorithms which induce and test very many concept expressions rely on performing coverage testing quickly, so the high complexity of instance checking by entailment over expressive DLs is an impediment to learning, as we cannot afford to re-classify the knowledge-base each time a new candidate expression is generated to determine its cover. Furthermore, coverage checking by entailment in an open-world will fail to indicate instance membership in concepts such as $\forall r.C$, $\leq^n r.C$ or $\neg C$ unless explicitly asserted against equivalent expressions in the ABox. However, inducing concepts with these constructs may be helpful and effective in characterising hypotheses.

One approach to address the complexity of coverage checking for concept learning is taken by the DL-LEARNER system which pre-classifies the knowledge-base (wrt. $\mathcal{I}$) with a reasoner. This computes the set of all named individuals in the interpretation of each concept and role in $\mathcal{K}$. While initially complex, the resulting completed ABox then contains all possible *known* assertions about atomic concepts and roles in the TBox, which reflects their $(\mathcal{I}, \mathcal{U})$ interpretation. The interpretation of any arbitrary $\mathcal{ALCCOQ}$ concept C can then be computed directly with respect to the definition of $(\mathcal{I}, \mathcal{U})$, which admits an exponential-time algorithm in the size of the expression and the size of the completed ABox. While not yet formally described, this technique has been referred to as Fast Instance Checks (FIC) in the DL-LEARNER system [10]. Basically, a check proceeds by testing if some named instance $a^{\mathcal{I}, \mathcal{U}} \in C^{\mathcal{I}, \mathcal{U}}$ by querying the ABox for named instances of sub-expressions of C in a top-down, depth-first backtracking manner until an interpretation is found which satisfies C . When incorporated

into a learning algorithm which generates concepts as candidates in a search, this method is known as *learning by interpretation*: we test if the known instances in the ABox are a model of C under $(\mathcal{I}, \mathcal{U})$. We will later show that this method of instance checking can be very fast in practice as it relies on a static database of instances for which re-classification of the knowledge-base is not required.

2.3 Inducing Concepts with Refinement Operators

To induce concept expressions in a knowledge-base, so-called *refinement operators* can be used to traverse the space of concepts which are ordered by the subsumption relation $\sqsubseteq$ (a partial order over concept expressions). Formally, if $(S, \sqsubseteq)$ is the ordered space of concepts, a downward (upward) refinement operator ρ is a mapping from S to 2^S such that for any $C \in S$ we have $C' \in \rho(C)$ implies $C' \sqsubseteq C$ ($C \sqsubseteq C'$). C' is called a *specialisation* (*generalisation*) of C . For example, concept C may be specialised with a refinement step which produces the expression $C \sqcap D$. Clearly, as $C \sqcap D \sqsubseteq C$, this is a downward refinement step. Other steps which provide downward (upward) refinements can be defined over the other elements of the hypothesis language. Detailed treatments of various refinement operators for expressive and inexpressive DLs and their properties have recently been explored by Lehmann and Hitzler [11].

The space of concept expressions induced by $\sqsubseteq_{(\mathcal{I}, \mathcal{U})}$ may be more constrained than the space induced by $\sqsubseteq_{\mathcal{I}}$. For example, consider the axiom $C \sqsubseteq_{\mathcal{I}} D \in \mathcal{T}$, where $C \equiv_{(\mathcal{I}, \mathcal{U})} D$. In this case, we can discard the concept C and prefer the more general D for the purposes of concept induction in a closed-world. Also, consider the case where $\mathcal{I} \not\models C \sqcap D \sqsubseteq \perp$ (i.e. C and D are not disjoint in the original knowledge-base). Despite this, the absence of any named individual as an instance of both C and D mean that they are disjoint under $(\mathcal{I}, \mathcal{U})$, and their conjunction $C \sqcap D$ should never be attempted by the refinement operator.

Prior to executing a learning algorithm, the subsumption relationships between atomic names in N_A and N_R entailed by $(\mathcal{I}, \mathcal{U})$ can be pre-computed to construct a constrained space in which to search for concepts using a refinement operator. For example, in any knowledge-base, there is a finite set M of expressions of the form $(A_0 \sqcap \dots \sqcap A_i)$ where $i \geq 0$, A are atomic concept names or single instance nominals $\{a\}$, and which is *minimal* where no two operands can be related by $\sqsubseteq_{(\mathcal{I}, \mathcal{U})}$. The set M captures concepts comprised of atomic terms which describe a subset of N_I . The set M of such concepts can also be limited to *localised sub-domains* as follows. The set of named example instances for the concept induction problem lie in the top-level *sub-domain* $\Delta_{\top}^{\mathcal{I}, \mathcal{U}} \subseteq \Delta^{\mathcal{I}, \mathcal{U}}$. A sub-domain Δ_{θ} captures the set of individuals from N_I in a locally confined context relative to some concept expression θ . For example, consider the expression $A \sqcap \exists r.(\top)$. The sub-domain $\Delta_{\exists r.-(A)}^{\mathcal{I}, \mathcal{U}} \subseteq \Delta^{\mathcal{I}, \mathcal{U}}$ captures the set of individuals of $\top$ which are r -successors of those in A . Localised sub-domains can be computed for any such expression relative to the closed-world interpretation and can be used as the basis for defining a *localised* set M_{θ} which captures only those atomic concept names which are pairwise non-redundant in conjunction to describe instances in the context of θ . This is useful to a refinement operator,

for example, in application to the sub-expression $\top$ in this example as $M_{r^-(A)}$ which captures the non-redundant set of concept expressions which can be used to specialise the expression.

For example, we may compute the set of localised sub-domains for every nested role path which is accessible from the example instances, such as $L = \{\Delta_\top, \Delta_{\exists r^-. \top}, \dots, \Delta_{\exists s^-. (\exists t^-. (... \top))}\}$. We can then compute the sets M_θ for every $\Delta_\theta \in L$, and use these when refining atomic concepts for each sub-expression in nested role expressions. Note that negated atomic concepts $(\neg A)^{\mathcal{I}, \mathcal{U}}$ for any $A \in M_\theta$ can also be defined locally as $\Delta_\theta^{\mathcal{I}, \mathcal{U}} \setminus A^{\mathcal{I}, \mathcal{U}}$, which can also be compared to other terms in M_θ with respect to $\sqsubseteq_{(\mathcal{I}, \mathcal{U})}$.

The concept expressions in M_θ , which we denote C^+ , are all the positively-defined minimal expressions capturing instances in the sub-domain of Δ_θ . The subsumption relation $\sqsubseteq_{(\mathcal{I}, \mathcal{U})}$ imposes a lattice structure relating the concepts $C^+ \in M_\theta$, and can be used to determine which atomic concepts in M_θ are subsumed, or overlap, any other concept C^+ . These atomic concepts can be combined with those concepts which they are subsumed by, or overlap with, as $C^+ \sqcap \neg A$. We define concepts of this kind as C^- , and describe them as being *maximal* if they are of the form $C^+ \sqcap (\prod \neg A_i)$ where no term $\neg A_i$ is redundant, and all possible such terms are included for each $C^+ \in M_\theta$. We denote the set of all maximal concepts C^- for some local sub-domain Δ_θ as M_θ^- , which is used when performing generalisation steps in the refinement search by excluding a single $\neg A$ subexpression from the conjunction. Following is an example of a downward (upward) refinement operator $\rho_\theta^\downarrow$ ($\rho_\theta^\uparrow$) which utilises this information for the concept language $\mathcal{ALCCQ}$:

$$\rho_\theta^\downarrow(\perp) = \emptyset \quad (1)$$

$$\rho_\theta^\downarrow(\top) = \{C_1^+ \sqcup \dots \sqcup C_m^+ \mid C_i^+ \in M_\theta, 1 \leq m \leq \max_\sqcup\} \quad (2)$$

$$\begin{aligned} \rho_\theta^\downarrow(\phi) = & \{\phi \sqcap \neg A \mid A \in M_\theta\} \cup \\ & \{\phi \sqcap \geq^r. (C_i^+) \mid C_i^+ \in M_{\exists r^-. \theta}, 1 \leq n \leq \max_{\theta, r}\} \cup \\ & \{\phi \sqcap \leq^r. (C_i^-) \mid C_i^- \in M_{\exists r^-. \theta}^-, 0 \leq n \leq \max_{\theta, r} - 1\} \cup \\ & \{\phi \sqcap \forall r. (C_i^+) \mid C_i^+ \in M_{\exists r^-. \theta}\} \end{aligned} \quad (3)$$

$$\rho_\theta^\downarrow(\sqcup \Phi) = \{\sqcup \Psi \mid \Psi = \Phi \setminus \{\phi_i\} \cup \rho_\theta^\downarrow(\phi_i), \phi_i \in \Phi\} \quad (4)$$

$$\rho_\theta^\downarrow(\forall r. (\phi)) = \{\forall r. (\psi) \mid \psi \in \rho_{\exists r^-. \theta}^\downarrow(\phi)\} \quad (5)$$

$$\rho_\theta^\downarrow(\geq^r. (\phi)) = \{\geq^r. (\psi) \mid \psi \in \rho_{\exists r^-. \theta}^\downarrow(\phi)\} \quad (6)$$

$$\rho_\theta^\downarrow(\leq^r. (\phi)) = \{\leq^r. (\psi) \mid \psi \in \rho_{\exists r^-. \theta}^\downarrow(\phi)\} \quad (7)$$

$$\rho_\theta^\uparrow(\perp) = \{C_i^- \mid C_i^- \in M_\theta^-\} \quad (8)$$

$$\rho_\theta^\uparrow(\prod \Phi) = \{\prod \Psi \mid \Psi = \Phi \setminus \{\neg A\}, \neg A \in \Phi\} \quad (9)$$

$$\rho_\theta^\uparrow(\phi) = \{\phi \sqcup \perp\} \quad (10)$$

where $\max_\sqcup$ is a pre-set maximum number of disjunct operands, $\max_{\theta, r}$ is the maximum number of any r -successors in context θ , and $\prod \Phi$ ($\sqcup \Phi$) refers to the conjunction (disjunction) of the set of expressions $\Phi = \{\phi_1, \dots, \phi_n\}$. Note that ρ generates expressions in *negation normal form* (nnf) where $\neg$ only appears in front of atomic concept names A as opposed to negations of arbitrary concepts ($\neg C$ which can always be transformed into equivalent nnf concepts). This refinement operator is *incomplete* in that certain $\mathcal{ALCCQ}$ concept expressions cannot

be reached. This is because we do not define role expressions nested within maximum quantified role expressions, as we do not prescribe an *upward* refinement method for such expressions, and the fact that expressions are globally bounded to a pre-set maximum number of disjuncts. The operator ρ also has the properties of *local finiteness* (all refinements of any concept are a finite set of new concepts); *improperness* in that refinements of any concept C may result in a semantically equivalent concept D where $C^{\mathcal{I},\mathcal{U}} = D^{\mathcal{I},\mathcal{U}}$, and *redundancy* in that multiple refinement chains of a single concept C may lead to semantically equivalent concepts. Of these properties, *properness* and *redundancy* most influence the efficiency of the operator to search the space of concepts. However, the refinement steps are largely confined to options which are non-redundant under $(\mathcal{I},\mathcal{U})$, thus reducing the amount of redundancy in the search.

2.4 Considerations for Concept Induction with $(\mathcal{I},\mathcal{U})$

Under the closed-world interpretation $(\mathcal{I},\mathcal{U})$, it is possible to induce a concept expression H which has a non-empty interpretation $H^{\mathcal{I},\mathcal{U}} \neq \emptyset$ but is actually unsatisfiable with respect to a knowledge-base in an open world ($H^{\mathcal{I}} = \emptyset$). For example, consider a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{A} = \{A(i), p(i, j), C(j)\}$ and $\mathcal{T} = \{A \sqsubseteq \geq^2 p.C\}$. Recall that an interpretation $(\mathcal{I},\mathcal{U})$ is a *model* of $\mathcal{T}$ if, for all axioms $C \sqsubseteq D \in \mathcal{T}$ we have $C^{\mathcal{I},\mathcal{U}} \subseteq D^{\mathcal{I},\mathcal{U}}$. In this example, $A^{\mathcal{I},\mathcal{U}} = \{i\}$ and $(\geq^2 p.C)^{\mathcal{I},\mathcal{U}} = \emptyset$ and therefore $(\mathcal{I},\mathcal{U})$ does not satisfy $\mathcal{T}$. The underlying problem here is that the derivation of the interpretation $(\geq^2 p.C)^{\mathcal{I},\mathcal{U}}$ relied exclusively on *known* information in $\mathcal{K}$. If information is *missing* from $\mathcal{A}$ with respect to $\mathcal{T}$, it may lead a learning algorithm relying on $(\mathcal{I},\mathcal{U})$ for computing hypothesis coverage to produce an expression which is unsatisfiable with respect to $\mathcal{T}$. In this case, one such expression is $(A \sqcap \leq^1 p.C)$ where $(A \sqcap \leq^1 p.C)^{\mathcal{I},\mathcal{U}} = \{i\}$, but clearly this expression is unsatisfiable with respect to $\mathcal{T}$ as $(A \sqcap \leq^1 p.C)^{\mathcal{I}} = \emptyset$.

A related problem occurs, for example, when the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{A} = \{P_1(i), P_2(j), p(i, k)\}$ and $\mathcal{T} = \{P_1 \sqsubseteq C, P_2 \sqsubseteq C, C \sqsubseteq \exists p.\top\}$ for the label classes P_1 and P_2 . In this case, we find that $(\exists p.\top)^{\mathcal{I}} = \{i^{\mathcal{I}}, j^{\mathcal{I}}, \dots\}$, however $(\exists p.\top)^{\mathcal{I},\mathcal{U}} = \{i\}$ as no p -successor of j is known in $\mathcal{A}$. From the perspective of a machine learner performing binary classification, the cover of hypothesis $\exists p.\top$ computed by $(\mathcal{I},\mathcal{U})$ appears to correctly describe all instances of partition P_1 and no instances of P_2 , and is thus considered a perfect characterisation of P_1 . Clearly however, with respect to $\mathcal{T}$, such a characterisation is incorrect.

The situation of inducing an unsatisfiable hypothesis with respect to $\mathcal{T}$ or mischaracterising its performance stems from data which is *missing in $\mathcal{A}$ with respect to $\mathcal{T}$* . In general, a hypothesis H (or a sub-expression thereof) which effectively discriminates examples (or parts thereof) with label P_0 from others in P_1 may only have been induced because data was present in the cover of H for examples in P_0 and was otherwise missing on all those in P_1 . It is the *distribution* of missing data which influences an induction method to generate H . However, it is difficult to generally characterise *which* missing data will influence a concept learner to generate problematic concepts, as this depends on the distribution of missing data and the learning strategy. Nevertheless, a method exists for

Algorithm 1. Beam Search for Top- k Concepts

```

1:  $F := \{\top\}$ 
2:  $S := \emptyset$ 
3: while  $\#F > 0$  and  $\#S < S_{\text{MAX}}$  do
4:    $h_{\text{max}} \in \arg \max_{h \in F} \text{util}(h)$  ▷ Arbitrary equal best hypothesis
5:    $F := F \setminus \{h_{\text{max}}\}$  ▷ Remove the chosen best from the frontier
6:   for all  $h' \in \rho_{\theta}^{\downarrow}(h_{\text{max}})$  do ▷ For each single-step refinement of  $h_{\text{max}}$ 
7:      $u_{h'} := \text{util}(h')$  ▷ Determine performance based on the cover
8:     if not  $\text{insuff}(u_{h'})$  and  $\forall s \in S, h' \sqsubseteq_{\mathcal{I}, \mathcal{U}} s \rightarrow u_{h'} > \text{util}(s)$  then
9:       if  $\text{suff}(u_{h'})$  then
10:         $S := S \cup \{h'\}$ 
11:       else
12:         $F := F \cup \{h'\}$ 
13:        if  $\#F > F_{\text{MAX}}$  then ▷ Beam width exceeded
14:           $h_{\text{min}} \in \arg \min_{h \in F} \text{util}(h)$  ▷ Arbitrary equal worst hypothesis
15:           $F := F \setminus \{h_{\text{min}}\}$ 
16:        end if
17:      end if
18:    end if
19:     $S_{\text{sub}} := \{s : s \in S, s \sqsubseteq_{\mathcal{I}, \mathcal{U}} h' \wedge \text{util}(s) \leq u_{h'}\}$ 
20:     $S := (S \cup \{h'\}) \setminus S_{\text{sub}}$ 
21:  end for
22: end while

```

recognising that data is missing in $\mathcal{K}$, which is to perform *integrity checking* of assertions in $\mathcal{A}$ against axioms in $\mathcal{T}$ [8]. This involves testing that, for each axiom $C \sqsubseteq D$ in $\mathcal{T}$, the condition $C^{\mathcal{I}, \mathcal{U}} \subseteq D^{\mathcal{I}, \mathcal{U}}$ holds². If it does not, instances in $C^{\mathcal{I}, \mathcal{U}}$ which are not in $D^{\mathcal{I}, \mathcal{U}}$ have missing data. Data missing from D could be handled by excluding any examples incorporating this data from the learning problem, or by entirely pruning sub-parts from all examples where data is missing so the expressions which may otherwise capture them are not relied upon in the learning problem. Alternatively, asserting new data which is detected to be missing may be possible [12].

We believe that the benefits of using $(\mathcal{I}, \mathcal{U})$ for efficient coverage testing far outweigh the drawbacks which include the possibility of generating expressions which violate axioms in $\mathcal{T}$. This is particularly the case when we aim to use concept induction to perform analyses over the data such as data mining for patterns, where concepts are induced to be human readable descriptions of interesting patterns in the database, and are not intended to be re-incorporated into the TBox.

2.5 Learning Procedure

Algorithm 1 is a general-purpose beam search for finding the top- k OWL class hypothesis expressions ranked by a utility function, util . Hypotheses are refined by $\rho_{\theta}^{\downarrow}$ and, if suitable, are stored in a frontier set F in descending order by utility,

² Possibly subject to expressivity restrictions on the background knowledge [8].

up to some maximum cardinality F_{MAX} (the beam width). The boolean functions *suff* (*insuff*) capture the conditions under which hypotheses can (never) be considered solutions. The definition of *util*, and thus which hypotheses are extracted first from F for refinement at any point in the algorithm, depends on the learning problem. For example, *util* may rank hypotheses based a function of their length (the number of symbols in the expression) together with some *quality* function, $qual(H)$. For example, in an unsupervised learning setting ($\mathbf{P} = \{P_1\}$), *relative frequency* (support) defined as $qual(H) = \frac{\#H^{\mathcal{I},\mathcal{U}}}{\#P_1}$ can be thresholded as $suff(H) = qual(H) \geq 0.75$ (hypotheses covering at least 75% of all examples) and where $insuff(H) = qual(H) < 0.75$. Care must be taken such that *insuff* does not force the rejection of hypotheses which may *increase* their value of *qual* with further refinements³. An example of such a quality function is *accuracy* and is often used in the setting of classification (e.g., $\mathbf{P} = \{P_1, P_2\}$ to describe P_1) as $acc(H) = \frac{|H^{\mathcal{I},\mathcal{U}} \cap P_1| + |P_2 \setminus H^{\mathcal{I},\mathcal{U}}|}{|P_1| + |P_2|}$. Here, we can set $suff(H) = acc(H) \geq 0.75$ (solutions have $\geq 75\%$ accuracy) and $insuff(H) = |H^{\mathcal{I},\mathcal{U}} \cap P_1| + |P_2| - 0.75(|P_1| + |P_2|) < |H^{\mathcal{I},\mathcal{U}} \cap P_2|$ (capturing the fact that no hypothesis covering less than a certain number of examples from P_1 can ever have an accuracy $\geq 75\%$). Line 8 of Algorithm 1 ensures that weak hypotheses which cover subsets of examples of relatively stronger and more general solutions are not considered. Similarly, on line 19, S_{sub} is computed to contain the set of solutions which are subsumed by a hypothesis with a utility at least as strong. Elements of S_{sub} are rejected from the solution set in light of the new hypothesis in this case. The algorithm terminates when at least S_{MAX} solutions are found or the frontier becomes empty, but could be terminated after a fixed time to inspect any current best solutions in S .

3 Implementation and Evaluation

We have tested Algorithm 1 with a Java application written using the OWL-API [13] and the Pellet OWL reasoner [14]. In our implementation, named individuals in local sub-domains are mapped to integers in the range $[0, |\Delta_\theta|)$ to permit efficient set operations when computing the $(\mathcal{I}, \mathcal{U})$ interpretation of concepts. Additionally, our implementation computes all entailments of the form $A_0 \sqsubseteq_{(\mathcal{I}, \mathcal{U})} A_1$ between atoms described in §2.3 to infer equivalence, subsumption and disjointness in each local sub-domain prior to executing the algorithm.

To measure the performance of our learning algorithm and refinement operator, we profiled our software using the carcinogenesis data set⁴ for the problem of binary classification. This data set consists of an OWL ontology with 145 classes, 19 properties, and around 22,000 individuals amounting to around 98,000 RDF triples. We tested our software on a 2.8GHz Intel-i7 machine with 8Gb RAM. Loading and classifying the knowledge base using Pellet took 20 seconds and a

³ In machine learning, such functions (including accuracy) are known as not having the property of *anti-monotonicity*.

⁴ Available from <http://dl-learner.org/wiki/Carcinogenesis>

Fig. 1. Plot of the number of hypotheses generated and tested against the best accuracy for the carcinogenesis problem⁴. Each point is labelled with the time in seconds when the new best concept was found. Note that Alg 1., CELOE and OCEL were terminated after 44 seconds, 600 seconds and 2000 seconds, respectively; see Figure 2 for details.

further 20 seconds to compute $(\mathcal{I}, \mathcal{U})$ entailments revealing the relationship of atoms under $(\mathcal{I}, \mathcal{U})$ in each local sub-domain (see §2.3 for more detail).

Alg. 1			OCEL			CELOE		
Accuracy	Searched	Time (s)	Accuracy	Searched	Time (s)	Accuracy	Searched	Time (s)
0.66	33,799	5.2	0.659	10,820	11.1	0.555	987	4.3
0.67	33,725	6.8	0.671	32,829	741	0.626	10,953	11.5
0.68	66,940	9.9	0.6766	42,939	1506	0.632	91,104	98.7
0.69	286,003	44.52	0.6795	48,689	2000	0.632	855,197	600

Fig. 2. Data for Figure 1 showing the time taken to arrive at concepts with particular accuracy after searching and testing a number of concept expressions

We observed that with our implementation, high-accuracy concepts of lengths between 10 and 15 were generated in less than one minute, the best concept being produced after 44.5 seconds with 69.9% accuracy with a fixed maximum false discovery rate of 0.32. We then executed the experiment with DL-LEARNER using each of the main class learning algorithms it implements, OCEL and CELOE, which generate concept expressions in the same DL language $\mathcal{ALCOQ}$. Our goal was to test the speed and efficiency of our method and implementation in finding high-accuracy concepts against these algorithms.

Our first test was with the OCEL algorithm which tested a total of 48,689 expressions in 2000 seconds, locating a concept with a highest accuracy of 67.95%. Secondly, we tried the CELOE algorithm which we let test 855,197 expressions in 600 seconds, locating a concept with a best accuracy of 63.2%. A (truncated) plot of the results are shown in Figure 1. These results show that, for this problem, our implementation significantly outperforms both OCEL and CELOE by locating high-accuracy concepts in much less time, as summarised in Figure 2. Each implementation utilised a single thread to search and test expressions for which coverage was computed using the FIC method (§2.2). We believe that our

implementation gained significant advantage from using the constraints implied by the $(\mathcal{I}, \mathcal{U})$ interpretation to speed up both the search for concepts but also their testing, as the interpretation permitted us to store the database in memory at runtime and not require any complex reasoner instance checks. We note that the under-performance of the CELOE algorithm in this test may be explained by the fact that it is geared towards searching for short concepts, whereas high-accuracy concepts for this problem appear to be long.

Note that, in our implementation, the quality function used by the main search algorithm was chosen to align with those used by OCEL and CELOE, which was to use accuracy, accuracy gain and a concept length penalty as $util(H) = acc(H) + 0.5(acc(H) - acc(H')) - 0.2 \cdot length(H)$, where $acc(H)$ is the accuracy of H over the full example data set (see §2.5) and H' is a (parent) concept which was refined in one step to reach H .

4 Conclusion and Future Work

We have presented a closed-world interpretation suitable for concept induction by refinement-based search methods in OWL knowledge bases which may be underpinned by highly expressive description logics. We have also described a refinement operator for the concept language $\mathcal{ALCOQ}$ which leverages the constraints imposed upon the space of concept expressions structured by $\sqsubseteq$ under the closed-world interpretation. We have also presented a general-purpose algorithm for concept induction suitable for a variety of data mining and machine learning tasks involving this refinement operator. Our test over a large dataset representing a benchmark classification problem (carcinogenesis) reveals that our methods and implementation are efficient and outperforms the DL-LEARNER system using either the OCEL or the CELOE class expression algorithms.

Current work includes developing our implementation specifically for performing redundancy-aware top- k pattern mining in parallel, for which the optimisations we have presented in this paper are especially desirable when analysing large knowledge bases using highly expressive DL hypothesis languages. We are also extending the refinement operator to efficiently handle concrete domains to account for OWL datatype expressions, primarily numerical range restrictions. We are aiming to release an implementation of our methods as a branch of the DL-LEARNER implementation which is open-source⁵. Finally, we are currently applying our methods to a large-scale problem in the field of structural biology called protein crystallisation [15,1] for which they were primarily designed.

References

1. Newman, J., Bolton, E.E., Müller-Dieckmann, J., Fazio, V.J., Gallagher, D.T., Lovell, D., Luft, J.R., Peat, T.S., Ratcliffe, D., Sayle, R.A., Snell, E.H., Taylor, K., Vallotton, P., Velanker, S., von Delft, F.: On the need for an international effort to capture, share and use crystallization screening data. *Acta Crystallographica Section F Structural Biology and Crystallization Communications* 68(3), 253–258 (2012)

⁵ DL-LEARNER project page: <http://aksw.org/Projects/DLLearner.html>

2. Klein, J., Jupp, S., Moulos, P., Fernandez, M., Buffin-Meyer, B., Casemayou, A., Chaaya, R., Charonis, A., Bascands, J.L., Stevens, R., Schanstra, J.P.: The KUPKB: a novel web application to access multiomics data on kidney disease. *FASEB Journal: official publication of the Federation of American Societies for Experimental Biology* 26(5), 2145–2153 (2012) PMID: 22345404
3. Lehmann, J.: DL-learner: Learning concepts in description logics. *Journal of Machine Learning Research* 10, 2639–2642 (2009)
4. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence* 26(2), 139–159 (2007)
5. Fanizzi, N., d’Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In: Železný, F., Lavrač, N. (eds.) *ILP 2008. LNCS (LNAI)*, vol. 5194, pp. 107–121. Springer, Heidelberg (2008)
6. Ławrynowicz, A., Potoniec, J.: Fr-ONT: An algorithm for frequent concept mining with formal ontologies. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) *ISMIS 2011. LNCS*, vol. 6804, pp. 428–437. Springer, Heidelberg (2011)
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, New York (2003)
8. Tao, J., Sirin, E., Bao, J., McGuinness, D.L.: Integrity constraints in owl. In: *AAAI2010: Proceedings of the AAAI Conference on Artificial Intelligence* (2010)
9. Kazakov, Y.: RIQ and SROIQ are harder than SHOIQ. In: *Proc. KR 2008* (2008)
10. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. *Semantics: Science, Services and Agents on the World Wide Web* 9(1), 71–81 (2011)
11. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. *Machine Learning* 78, 203–250 (2010)
12. Tao, J.: *Integrity Constraints for the Semantic Web: An OWL2-DL Extension*. PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, USA, AAI3530046 (2012)
13. Horridge, M., Bechhofer, S.: The OWL API: a java API for OWL ontologies. *Semantic Web* 2(1), 11–21 (2011)
14. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2), 51–53 (2007)
15. Ratcliffe, D., Taylor, K., Newman, J.: Ontology-based machine learning for protein crystallisation. In: *Australasian Ontology Workshop (AOW 2011)*, Perth, Australia, vol. 132, CRPIT (2011)