Concept-Based Semantic Difference in Expressive Description Logics

Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler

School of Computer Science
University of Manchester
Manchester, United Kingdom

Abstract. Detecting, much less understanding, the difference between two description logic based ontologies is challenging for ontology engineers due, in part, to the possibility of complex, non-local logic effects of axiom changes. It is often quite difficult to even determine which terms have had their meaning altered by a change. To address this, various principled notions of "semantic diff" (based on deductive inseparability) have been proposed in the literature and have been shown to be computationally practical for the expressively restricted case of $\mathcal{ELH}^r$-terminologies (which covers significant fragments of SNOMED-CT). However, problems arise even for such limited logics as $\mathcal{ALC}$: First, computation gets more difficult, becoming undecidable for logics such as $\mathcal{SROIQ}$ which underly the Web Ontology Language (OWL). Second, the presence of negation and disjunction make the standard semantic difference too sensitive to change: essentially, any logically effectual change always affects all terms in the ontology. To address these issues, we formulate the central notion of finding the *minimal change set* based on model inseparability, and present a method to differentiate changes which are specific to (and "of interest" for) particular concept names. Subsequently we present a series of computable approximations, and compare the variously approximated change sets over a series of versions of the NCI Thesaurus (NCIt).

1 Introduction

Determining the significant differences between two documents (so-called "diff") is a standard and significant problem across a wide range of activities, notably software development. Standard textual diffing algorithms perform poorly on description logic (DL) based ontologies, both for structural reasons (e.g., ontology serializations, such as those of OWL, tend not to impose stable ordering of axioms), and due to the highly non-local and unintuitive logical effects of changes to axioms. Syntactic diffs, such as those based on OWL's notion of "structural equivalence" [4, 8, 12], detect axiomatic changes between ontologies, but fall short on the identification of differences w.r.t. their entailment sets. Recent notions of semantic difference based on conservative extensions have provided a robust theoretical and practical basis for analysing these logical effects. In particular, they provide a means for determining which terms have had their meaning "affected" by an edit even if that effect is not readily determined by syntactic analysis.

Unfortunately, semantic difference is computationally expensive even for inexpressive logics such as $\mathcal{EL}$. For the very expressive logics such as $\mathcal{SROIQ}$ (the DL underlying OWL 2) it is undecidable [10]. Furthermore, as we discuss in this paper, semantic difference runs into other difficulties in more expressive logics. In particular, if we compare entailment sets over logics with disjunction and negation we easily end up with vacuously altered terms: any logically effectual change will alter the meaning of every term.

In this paper, we provide a non-trivializable notion of semantic difference and a series of computable approximations of it for expressive description logics. We evaluate these algorithms on a select subset of the National Cancer Institute Thesaurus (NCIt) corpus, comparing the changes found via the proposed approximations and related approaches. Our experiments show that one approximation, "Grammar diff", finds significantly more changes than all other methods across the corpus and far more than are identified in the NCIt change logs.

2 Preliminaries

We assume the reader to be reasonably familiar with ontologies and OWL, as well as the underlying description logics (DLs) [1]. We use *terms* to refer to concept and role names. When comparing two ontologies we refer to them as $\mathcal{O}_1$ and $\mathcal{O}_2$, and their *signatures* (i.e., the set of terms occurring in them) as $\widetilde{\mathcal{O}}_1$ and $\widetilde{\mathcal{O}}_2$, respectively. The signature of an axiom α is denoted $\widetilde{\alpha}$. Throughout this paper we use the standard description and first order logic notion of entailment; an axiom α entailed by an ontology $\mathcal{O}$ is denoted $\mathcal{O} \models \alpha$. We refer to an *effectual* addition (removal) from $\mathcal{O}_1$ to $\mathcal{O}_2$ as an axiom α such that $\alpha \in \mathcal{O}_2$ and $\mathcal{O}_1 \not\models \alpha$ ($\alpha \in \mathcal{O}_1$ and $\mathcal{O}_2 \not\models \alpha$) [4]. Thus two ontologies are logically equivalent, denoted $\mathcal{O}_1 \equiv \mathcal{O}_2$, if there is no effectual change (addition or removal) between $\mathcal{O}_1$ and $\mathcal{O}_2$. We also use the notion of a *locality-based module* [2]; a module $\mathcal{M}$ of $\mathcal{O}$ for a set of terms (signature) Σ is a subset of $\mathcal{O}$ that preserves all entailments of $\mathcal{O}$ w.r.t. Σ. A $\bot$-*module* ($\top$-*module*) extracted from an ontology $\mathcal{O}$ for Σ is denoted $\bot$-*mod*$(\Sigma, \mathcal{O})$ ($\top$-*mod*$(\Sigma, \mathcal{O})$). The set of *subconcepts* of an ontology $\mathcal{O}$ is recursively defined as all subconcepts found in each axiom of $\mathcal{O}$, plus $\{\top, \bot\}$.

The restriction of an interpretation $\mathcal{I}$ to a set of terms Σ is denoted $\mathcal{I}|_\Sigma$. Two interpretations $\mathcal{I}$ and $\mathcal{J}$ coincide on a signature Σ (denoted $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$) if $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $t^\mathcal{I} = t^\mathcal{J}$ for each $t \in \Sigma$.

Throughout this paper we use the notion of model conservative extension (mCE) [3, 10], and associated inseparability relation [14]. The notions of mCE-based inseparability, Σ-difference and Σ-entailment are, respectively:

Definition 1 *Given two ontologies $\mathcal{O}_1$, $\mathcal{O}_2$ over a DL $\mathcal{L}$, and a signature Σ.*

(1) $\mathcal{O}_2$ *is model Σ-inseparable from* $\mathcal{O}_1$ *($\mathcal{O}_1 \equiv_\Sigma^{mCE} \mathcal{O}_2$) w.r.t. $\mathcal{L}$ if* $\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{O}_2\}$

(2) $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma = \{\eta \mid \mathcal{O}_1 \not\models \eta, \mathcal{O}_2 \models \eta$ *and η is a GCI over $\mathcal{L}$, with* $\widetilde{\eta} \subseteq \Sigma\}$

(3) $\mathcal{O}_1$ *Σ-entails* $\mathcal{O}_2$ *if* $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma = \emptyset$

3 State of the Art in Semantic Diff

The tool ContentCVS [6] employs a notion of deductive difference (for OWL 2 ontologies) which takes into account entailments of type $A \sqsubseteq C$,[1] where C is a concept formed over grammar G_{cvs} and A, B are concept names, as follows:

Grammar G_{cvs}

$C \longrightarrow B \mid \exists r.B \mid \forall r.B \mid \neg B$

The rationale behind the use of this grammar is not exactly clear, and seems rather ad hoc. In a user study of ContentCVS, users criticised "the excessive amount of information displayed when using larger approximations of the deductive difference" [6]. This suggests that, instead of focusing on presenting entailments in the difference, we might prefer to present which concept names are affected by those entailments, and how (e.g., specialised or generalised).

The diff method underlying the system CEX [7] establishes a way to compute the semantic differences between two ontologies,[2] based on the notion of Σ-entailment, and corresponding diff notion Σ-difference. The output of CEX is a set of entailed axioms in the Σ-difference, so called *witness axioms*, and associated affected terms (denoted $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$). The set $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ contains specialised (denoted $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{L}}_\Sigma$) and generalised ($\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{R}}_\Sigma$) concept names, as defined in [7]. The set $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{L}}_\Sigma$ contains those concept names A for which there is a witness axiom $\alpha : A \sqsubseteq C$ that follows from $\mathcal{O}_2$ but not $\mathcal{O}_1$. The concept C in such axioms α is called a *witness* for the change in A. In $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{R}}_\Sigma$ the witness is the subsumer rather than the subsumee.

The computational complexity of deciding Σ-entailment is undecidable for expressive DLs such as $\mathcal{SROIQ}$. For $\mathcal{EL}$ it is already ExpTime-complete [11], while for $\mathcal{ALC}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$ it is 2ExpTime-complete [10]. Aside from the high complexity result, a direct extension of Σ-difference for more expressive logics such as $\mathcal{ALC}$ would fail; when we step beyond $\mathcal{EL}$ as a witness language into more expressive logics with disjunction and negation, then we can create a vacuously true witness that would make $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ contain all terms in Σ (so long as $\mathcal{O}_1 \not\equiv \mathcal{O}_2$). The ontologies need not be in the witness language; in fact consider the following $\mathcal{EL}$ ontologies: $\mathcal{O}_1 = \{A \sqsubseteq B, C \sqsubseteq \top, D \sqsubseteq \top\}$, and $\mathcal{O}_2 = \{A \sqsubseteq B, C \sqsubseteq D\}$. Clearly $\mathcal{O}_2$ is a conservative extension of $\mathcal{O}_1$ w.r.t. $\Sigma = \{A, B\}$, but if we take $\Sigma' = \{\widetilde{\mathcal{O}_1} \cap \widetilde{\mathcal{O}_2}\}$ then that is no longer the case. A witness axiom for the separability would be, e.g., $\eta := A \sqsubseteq \neg C \sqcup D$. This witness "witnesses" a change to every concept $A' \in \Sigma'$; for each witness axiom $\eta' : A' \sqsubseteq \neg C \sqcup D$ we have that $\mathcal{O}_1 \not\models \eta'$, while $\mathcal{O}_2 \models \eta'$. Such a witness would suffice to pinpoint, according to Σ-difference, that all terms in Σ' have changed: $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma'} = \Sigma'$ since $\top \sqsubseteq \neg C \sqcup D$. Consequently, this kind of witnesses are uninteresting for any particular concept aside from $\top$. Likewise, a change $A \sqsubseteq \bot$ implies that, for all B in the signature of the ontology in question, we have that $A \sqsubseteq B$. Yet these consequences are of no interest to any concept B.

[1] Additionally, ContentCVS also compares role hierarchies.

[2] Albeit the implementation is restricted to acyclic $\mathcal{ELH}^r$ terminologies ($\mathcal{EL}$ extended with role inclusions and range restrictions).

Similar to the case of the least common subsumer [9], the presence of disjunction (and negation) trivialises definitions that are meaningful in less expressive logics. This phenomenon conveys the need to move to another diff notion when dealing with propositionally closed ontologies, one which distinguishes directly affected terms (thus "specific" changes) and indirectly affected terms (such as those via $\top$ and $\bot$ from previous examples).

4 Semantic Diff

Given the shortcomings of existing methodologies, we present a semantic diff method that *a)* determines which concepts have been affected by changes. For exposition reasons, we concentrate on concepts, though roles are easily added. And *b)* identifies which concepts have been directly (or indirectly) changed.

Ideally, a solution to these problems would be *1)* a computationally feasible function (for OWL 2 ontologies), *2)* based on a principled grammar, that *3)* returns those concept names affected by changes between two ontologies, while *4)* distinguishing whether each concept name is directly (or indirectly) specialised and/or generalised.

4.1 Determining the Change Set

Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, such that $\mathcal{O}_1 \not\equiv \mathcal{O}_2$ (i.e. there exists at least one effectual change in $\text{Diff}(\mathcal{O}_1, \mathcal{O}_2)$), we know that $\mathcal{O}_1$ and $\mathcal{O}_2$ are not Σ-inseparable (for $\Sigma = \widetilde{\mathcal{O}}_1 \cup \widetilde{\mathcal{O}}_2$) w.r.t. model inseparability, i.e. $\mathcal{O}_1 \not\equiv^{mCE}_{\Sigma} \mathcal{O}_2$ since an effectual change implies some change in semantics. In order to pinpoint this change, we need to find the set of terms Σ' s.t. $\mathcal{O}_1$ is mCE-inseparable from $\mathcal{O}_2$ w.r.t. the remaining signature $\Sigma \setminus \Sigma'$: $\mathcal{O}_1 \equiv^{mCE}_{\Sigma \setminus \Sigma'} \mathcal{O}_2$. Then we know that, from $\mathcal{O}_1$ to $\mathcal{O}_2$, there are no changes in entailments over $\Sigma \setminus \Sigma'$. We refer to this set of terms Σ' as the Minimal Change Set (denoted $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$), in the sense that we can formulate a non-trivial entailment η over Σ' s.t. $\mathcal{O}_1 \not\models \eta$ but $\mathcal{O}_2 \models \eta$. Thus we denote these terms as *affected*.

Definition 2 (Minimal Affected Terms) *A set $\Sigma' \subseteq \Sigma$ is a set of minimal affected terms between $\mathcal{O}_1$ and $\mathcal{O}_2$ if:*

$$\mathcal{O}_1 \not\equiv^{mCE}_{\Sigma'} \mathcal{O}_2 \text{ and for all } \Sigma'' \subsetneq \Sigma' : \mathcal{O}_1 \equiv^{mCE}_{\Sigma''} \mathcal{O}_2.$$

The set of all such sets is denoted $\text{MinAT}(\mathcal{O}_1, \mathcal{O}_2)$.

In order to form the minimal change set, we take the union over all sets of affected terms in $\text{MinAT}(\mathcal{O}_1, \mathcal{O}_2)$.

Definition 3 (Minimal Change Set) *The minimal change set, denoted* $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$*, of two ontologies is defined as follows:*

$$\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2) := \bigcup \text{MinAT}(\mathcal{O}_1, \mathcal{O}_2)\,.$$

Given a set of witness axioms, we can tell apart specialised and generalised concepts depending on whether the witness concept is on the right hand side (RHS) or the left hand side (LHS) of the witness axiom, accordingly. Furthermore, we regard a concept name A as directly specialised (generalised) via some witness C if there is no concept name B that is a superconcept (subconcept) of A, and C is also a witness for a change in B. Otherwise A changed indirectly.

Definition 4 (Affected Terms) *For a diff function Φ, the sets of affected concept names for a signature Σ are:*

$$\Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\mathrm{L}}_{\Sigma} = \{A \in \Sigma \mid \textit{there exists } A \sqsubseteq C \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)_{\Sigma}\}$$

$$\Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\mathrm{R}}_{\Sigma} = \{A \in \Sigma \mid \textit{ there exists } C \sqsubseteq A \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)_{\Sigma}\}$$

$$\Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\top}_{\Sigma} = \begin{cases} \{\top\} & \textit{if there is a } \top \sqsubseteq C \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)_{\Sigma} \\ \emptyset & \textit{otherwise} \end{cases}$$

$$\Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\bot}_{\Sigma} = \begin{cases} \{\bot\} & \textit{if there is a } C \sqsubseteq \bot \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)_{\Sigma} \\ \emptyset & \textit{otherwise} \end{cases}$$

$$\Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)_{\Sigma} = \textstyle\bigcup_{Y \in \{L,R,\top,\bot\}} \Phi\text{-}\,\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{Y}_{\Sigma}$$

Given a concept name $A \in \Phi\text{-}\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\mathrm{L}}_{\Sigma}$ (analogously $A \in \Phi\text{-}\mathrm{AT}(\mathcal{O}_1,\mathcal{O}_2)^{\mathrm{R}}_{\Sigma}$), and a set of terms $\Sigma^+ := \Sigma \cup \{\top,\bot\}$:

A direct change of A is a witness C s.t. $A \sqsubseteq C$ $(C \sqsubseteq A) \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)$ and there is no $B \in \Sigma^+$ s.t. $\mathcal{O}_2 \models A \sqsubseteq B$ $(\mathcal{O}_2 \models B \sqsubseteq A)$, $\mathcal{O}_2 \not\models A \equiv B$, and $B \sqsubseteq C$ $(C \sqsubseteq B) \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)$.

An indirect change of A is a witness C s.t. $A \sqsubseteq C$ $(C \sqsubseteq A) \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)$ and there is at least one $B \in \Sigma^+$ s.t. $\mathcal{O}_2 \models A \sqsubseteq B$ $(\mathcal{O}_2 \models B \sqsubseteq A)$, $\mathcal{O}_2 \not\models A \equiv B$ and $B \sqsubseteq C$ $(C \sqsubseteq B) \in \Phi\text{-}\,\mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)$.

Concept A is purely directly changed if it is only directly changed (analogously for purely indirectly changed).

As an example, given ontologies $\mathcal{O}_1 := \{A \sqsubseteq B, \exists r.C \sqsubseteq D\}$ and $\mathcal{O}_2 := \mathcal{O}_1 \cup \{B \sqsubseteq \exists r.C\}$, we have that B is purely directly specialised via witness $\exists r.C$, while A is indirectly specialised via the same witness, since $\mathcal{O}_2 \models A \sqsubseteq B$ and $B \sqsubseteq \exists r.C \in \mathrm{Diff}(\mathcal{O}_1,\mathcal{O}_2)$, in other words, concept A changes via B.

The distinction between directly- and indirectly-affected concept names, and the separation of concepts affected via $\top$ and $\bot$, allows us to overcome the problems described in Section 3, w.r.t. propositionally closed description logics.

4.2 Computation

Deciding the minimal change set between two ontologies involves deciding whether, for a given signature Σ, two ontologies are mCE-inseparable w.r.t. Σ. Since mCE-inseparability is undecidable for $\mathcal{SROIQ}$ [10], we present two sound but incomplete approximations to the problem of computing the minimal change set: "Subconcept" and "Grammar" diffs.

In addition, and in order to provide a basis for comparison between diff notions, we define the set of differences which would be captured by a comparison of the concept hierarchies between two ontologies, i.e. differences in atomic subsumptions, as $\text{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$. Hereafter we refer to ContentCVS's diff notion as $\text{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$.

The first approximation, Subconcept diff (denoted $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$), is based on subconcepts of ontologies, wherein we check whether there is a difference in entailments of type $C \sqsubseteq D$, where C or D is a possibly complex concept from the set of Σ-subconcepts of $\mathcal{O}_1$ and $\mathcal{O}_2$ (see Definition 5). It is at least conceivable that many entailments will involve subconcepts, and, if that is the case, those would be witnesses that the user could understand, since they are explicitly asserted in either ontology. Moreover, this notion may exhibit entailment differences which would not show up if we restrict ourselves to either atomic subsumptions, or specific forms of entailments (in the manner of ContentCVS). The restriction to forms of concepts explicit in either ontology limits the amount of change captured. E.g., if we have $\mathcal{O}_1 = \{A \sqsubseteq \exists r.B\}$, and in $\mathcal{O}_2$ add an axiom $B \sqsubseteq \exists s.C$, the change $A \sqsubseteq \exists r.\exists s.C$ would not be found. However, the rationale behind this approach is that we could detect other kinds of change in a principled and relatively cheap way, e.g., $\mathcal{O}_1 = \{A \sqsubseteq B\}$, $\mathcal{O}_2 = \mathcal{O}_1 \cup \{B \sqsubseteq \exists r.(C \sqcap \exists r.D)\}$; we have that $\mathcal{O}_1 \not\models \alpha := A \sqsubseteq \exists r.(C \sqcap \exists r.D)$, while $\mathcal{O}_2 \models \alpha$.

In order to avoid only considering witnesses in their explicitly asserted form, we extend the previous diff notion and present Grammar diff (denoted $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$), which detects differences in additional types of entailments; the grammars below define the types of concepts taken into account by Grammar diff, where SC stands for a subconcept of $\mathcal{O}_1 \cup \mathcal{O}_2$.

Grammar G_L
$C \longrightarrow SC \mid SC \sqcup SC \mid \exists r.SC \mid \forall r.SC \mid \neg SC$
Grammar G_R
$C \longrightarrow SC \mid SC \sqcap SC \mid \exists r.SC \mid \forall r.SC \mid \neg SC$

The semantic difference between ontologies w.r.t. each mentioned diff notion is defined as follows:

Definition 5 *Given two ontologies and a signature Σ, the set of Σ-differences for a diff notion Φ is:*

$$\Phi\text{-}\,\text{Diff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma} := \{\eta \in \Phi\text{-}ax \mid \mathcal{O}_1 \not\models \eta \wedge \mathcal{O}_2 \models \eta \wedge \widetilde{\eta} \subseteq \Sigma\}$$

where the set Φ-ax is defined as follows:

$$\begin{array}{ll}
\textit{if } \Phi = At, & \{C \sqsubseteq D \mid C, D \in \Sigma\} \\
\textit{if } \Phi = Sub, & \{C \sqsubseteq D \mid C, D \textit{ subconcepts in } \mathcal{O}_1 \cup \mathcal{O}_2\} \\
\textit{if } \Phi = Gr, & \{C \sqsubseteq D \mid D \textit{ a concept over } G_L, \textit{or } C \textit{ a concept over } G_R\} \\
\textit{if } \Phi = Cvs, & \{C \sqsubseteq D \mid C \in \Sigma \textit{ and } D \textit{ a concept over } G_{cvs}\} \\
\textit{if } \Phi = CEX, & \{C \sqsubseteq D \mid C, D \textit{ subconcepts in } \mathcal{L}(\Sigma)\}
\end{array}$$

It is not hard to see that there are subset relations between each diff and the actual $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$ that they approximate, as per Lemma 1:

Lemma 1 *Given two ontologies and a signature Σ:*

$$\text{AtDiff-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{SubDiff-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{GrDiff-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$$
$$\text{CvsDiff-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{GrDiff-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$$

The current implementation of CEX only takes as input acyclic $\mathcal{ELH}^r$ *terminologies*, that is, $\mathcal{ELH}^r$ TBoxes which are 1) acyclic and 2) every concept appears (alone) on the left-hand side of an axiom exactly once. In order to apply CEX to knowledge bases that are more expressive than $\mathcal{ELH}^r$ terminologies, we rely on an approximation that uses CEX as a sub-routine.

Definition 6 (Approx-CEX) *Given two non-$\mathcal{ELH}^r$ ontologies, the Approx-CEX procedure is:*

1. *For $i \in \{1, 2\}$, approximate $\mathcal{O}_i$ as an $\mathcal{ELH}^r$ terminology, resulting in $\mathcal{O}'_i$:*
   (a) *Remove all non-$\mathcal{EL}$ axioms.*
   (b) *Break cycles (non-deterministically).*
   (c) *Remove all but one axiom with a given atomic left-hand side.*
2. *Apply CEX to $\mathcal{O}'_1$, $\mathcal{O}'_2$, resulting in a temporary change set:* TempCS.
3. *For each $\alpha \in TempCS$, add α to* FinalCS *if $\mathcal{O}_1 \not\models \alpha$ and $\mathcal{O}_2 \models \alpha$.*
4. *Return* FinalCS*; the set of axioms in the diff.*

Note that step 1 is parameterizable with any $\mathcal{ELH}^r$ approximation algorithm. Additionally, step 2 can be replaced with a diff implementation for more expressive logics, with either the input approximation (step 1) and soundness check (step 3) removed, or with an altered step 1 depending on the expressivity of the input. Step 4 in Definition 6 is necessary to ensure that changes detected within the $\mathcal{ELH}^r$ approximations (obtained in step 1) are sound changes w.r.t. the whole ontologies. Obviously, this approximation-based procedure throws away a lot of information and is not deterministic. However, even such an approximation can offer useful insight, esp. if it finds changes that other methods do not. There are more elaborate existing approximation approaches (e.g., [13]), but they generally do not produce $\mathcal{ELH}^r$ terminology, so their use requires either changing the approximation output or updating CEX to take non-terminological $\mathcal{EL}$ input.

5 Empirical Results

The object of our evaluation is a subset of the NCIt corpus used in [4], with expressivity ranging from $\mathcal{ALCH}(\mathcal{D})$ to $\mathcal{SH}(\mathcal{D})$. More specifically, we take into account 12 versions of the NCIt which contain concept-based change logs. In order to investigate the applicability of our approach we *(1)* compare the results obtained via our approximations with those output by Approx-CEX and ContentCVS, and *(2)* inspect whether the devised approximations capture any direct changes not reported in the NCIt change logs.

The experiment machine used is an Intel Xeon Quad-Core 3.20GHz, with 16Gb DDR3 RAM. The system runs Mac OS X 10.6.8, Java Virtual Machine (JVM v1.5), and all tests were run using the OWL API (v3.2.4) [5].[3]

In terms of computation times, on average computing $\text{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ takes ≈20 seconds, Approx-CEX takes ≈9 minutes, while computing $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ takes ≈35 minutes. The computation of $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ takes ≈14 hours for a subset of the ontology signature of size ≈1800 concept names, and ContentCVS ≈10 hours on the same randomly selected signature as $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$.[4]

5.1 Diff Comparison

The comparison of each diff w.r.t. number of affected concept names found is shown in Table 1, which displays the number of specialised concepts (L-changes), generalised concepts (R-changes), and the total number of affected concepts. Figure 1 shows a comparison of the number of affected concept names found by ContentCVS and Grammar diff within the randomly selected signatures. Note that, at this point, no distinction is made between direct and indirect changes.

Table 1: Number of affected concept names found by each diff, and their respective coverage w.r.t. affected concepts found by GrammarDiff.

NCIt	Approx-CEX			AtDiff			SubconceptDiff			GrammarDiff		
	L	R	Total	L	R	Total	L	R	Total	L	R	Total
1 (05.07d)	454	307	668	979	486	1,416	1,701	490	2,131	10,501	3,597	12,178
2 (05.10e)	413	648	851	792	499	1,208	1,436	518	1,816	11,366	3,442	12,975
3 (05.11f)	3,508	2,089	5,013	5,233	1,172	6,135	5,910	1,178	6,528	12,379	6,806	17,542
4 (05.12f)	1,400	2,813	2,950	2,358	1,485	3,676	45,825	1,495	45,932	19,547	13,691	28,305
5 (06.01c)	7,305	2,495	8,692	3,808	1,321	4,978	15,254	1,498	15,691	36,333	20,137	39,491
6 (06.02d)	1,131	684	1,520	3,502	624	3,923	5,806	663	6,203	10,621	11,331	19,741
7 (06.03d)	1,721	2,434	3,052	2,462	1,127	3,217	5,777	1,201	6,330	20,620	9,799	24,567
8 (06.04d)	417	1,382	1,590	6,284	1,631	6,806	6,952	1,674	7,428	10,275	7,576	14,047
9 (06.05d)	1,095	1,455	1,711	2,224	678	2,745	4,928	737	5,329	13,291	9,223	13,819
10 (06.06e)	1,649	1,002	2,154	4,073	607	4,553	5,992	663	6,415	26,161	5,345	28,005
11 (06.08d)	624	968	1,099	1,240	610	1,714	3,910	731	4,325	37,674	3,630	38,502
Avg. Cov.	9%	18%	12%	20%	12%	18%	52%	13%	41%			
Min. Cov.	2%	6%	3%	3%	6%	4%	10%	6%	11%			
Max. Cov.	28%	31%	29%	61%	22%	48%	100%	22%	100%			

Due to computational issues regarding Grammar diff and ContentCVS, instead of comparing each pair of NCIt versions w.r.t. $\Sigma = \widetilde{\mathcal{O}}_1 \cup \widetilde{\mathcal{O}}_2$ we take a random sample of the terms in the ontology (generally n ≈ 1800) such that a straightforward extrapolation allows us to determine that the true proportion of changed terms lies in the confidence interval (+-3%) with a 99% confidence level. In general, Grammar diff, even taking into account the confidence interval, consistently detects more changes (both L and R) than all other diffs. Also,

[3] `http://owlapi.sourceforge.net/`

[4] Note that, originally, ContentCVS only computes $\text{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\text{L}}_{\Sigma}$, but in order to provide a direct comparison with the diffs here proposed we also compute $\text{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\text{R}}_{\Sigma}$ according to ContentCVS's grammar.

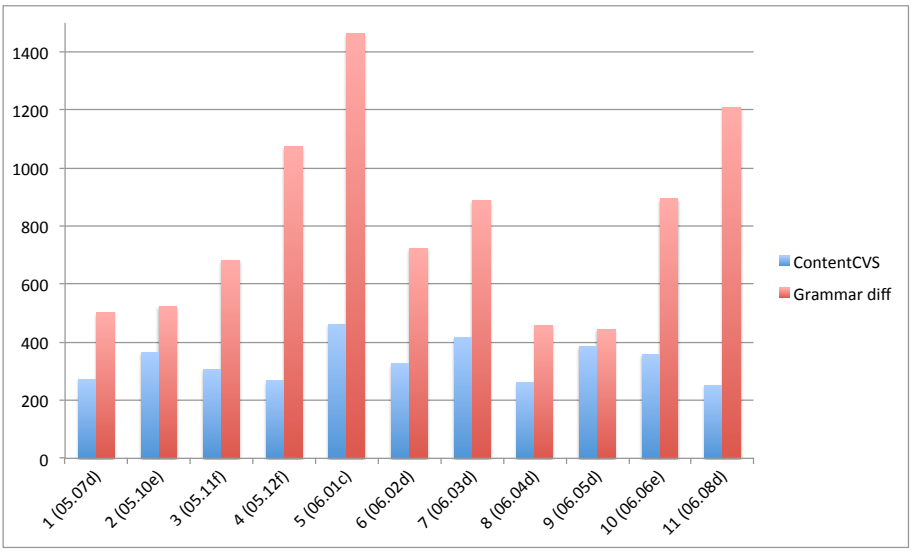

Fig. 1: Comparison of total number of affected concepts found by ContentCVS and Grammar diff (y-axis: number of concept names, x-axis: NCIt version).

despite the one case where the lower bound of detected changes is inferior to another diff, in version 4, it cannot be worse than Subconcept diff by Lemma 1.

5.2 Direct Changes in the NCIt Logs

The change logs supplied with each version of the NCIt contain those concept names which were subject to changes. However, it is unclear whether each reported change also (or solely) relates to annotation changes. It could be the case that a reported concept change is purely ineffectual. In spite of this ambiguity, it should be expected that a change log contains concept names that were directly changed, and this is what we aim to find out in our next experiment; we extract the concept names mentioned in the change log, and verify whether the obtained direct changes for each NCIt version are contained in said change logs. The results are shown in Table 2, where the affected concept names shown in Section 5.1 are partitioned into purely direct, purely indirect, or both directly and indirectly changed concepts. Overall, we see that the change logs do miss a lot of direct changes, more specifically, on average, $\text{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ reveals 767 changed concept names not mentioned in the change logs, while $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ uncovers 908 such concept names per NCIt version.

6 Discussion

First thing to notice is that SubDiff finds many more changes than AtDiff and Approx-CEX, while often not reaching close to the projected values of GrammarDiff (the average coverage being 41%). The latter, as expected, captures far more changes within the selected signatures than ContentCVS.

Table 2: Number of purely direct (P.D.), purely indirect (P.I.), and both directly and indirectly (Mix) changed concepts. Number of directly changed concepts that do not appear in the NCIt change logs (denoted Missed).

NCIt	AtDiff							SubDiff						
	L			R			Missed	L			R			Missed
	Mix	P.D.	P.I.	Mix	P.D.	P.I.		Mix	P.D.	P.I.	Mix	P.D.	P.I.	
1	524	122	333	88	206	192	798	686	134	881	88	210	192	953
2	440	125	227	95	179	225	149	803	344	289	96	198	224	211
3	2,106	215	2,912	211	680	281	315	2,242	549	3,119	212	686	280	445
4	1,498	126	734	146	1,041	298	190	2,647	78	43,100	148	1,050	297	432
5	1,401	154	2,253	127	882	312	243	6,511	1,527	7,216	304	882	312	317
6	813	77	2,612	153	232	239	199	1,163	143	4,500	161	240	262	199
7	984	206	1,272	256	448	423	273	2,400	320	3,057	267	513	421	511
8	5,923	152	209	154	1,267	210	5,546	5930	483	539	157	1,308	209	5,723
9	870	611	743	166	254	258	207	1,775	832	2,321	171	307	259	322
10	594	2,727	752	145	225	237	216	2,110	2,854	1,028	147	280	236	298
11	586	167	487	139	239	232	300	1,050	354	2,506	147	325	259	582

Considering the high number of affected concepts found by SubDiff in versions 4 and 5 of the NCIt, one can argue that analysing such a change set would be rather unpleasant. By categorising concept names in the change set according to whether they are directly or indirectly affected, we can greatly reduce the information overload; notice that, e.g., in version 4 there are 45,825 specialised concepts, out of which there are only 78 purely directly changed concepts, and the majority of the remainder are purely indirect changes (43,100). Similarly in version 5, from 15,254 specialised concepts there are only 1,527 purely direct changes. Immediately we see that this mechanism can provide an especially helpful means to assist change analysis, by, e.g., confining the changes shown upfront to only those which are (purely) direct.

Despite the optimisations applied in GrammarDiff's implementation, e.g., for $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)^L_\Sigma$ we start by verifying whether there exists some effectual change between $\bot$-mod$(\{A\}, \mathcal{O}_1)$ and $\bot$-mod$(\{A\}, \mathcal{O}_2)$, for each $A \in \Sigma$ (analogously we use $\top$-modules for $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)^R_\Sigma$), only considering witnesses whose signature is contained in the module signature, stopping once we find a single witness for a concept name, the computation of $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ still takes long, and needs further optimisations. The major bottleneck is that the $\top$-modules for a concept name provide too big an approximation, e.g., for a top-level concept its $\top$-module contains almost the whole ontology. Thus $\top$-modules do not restrict much of our search space, not at least in the same way as $\bot$-modules do. Additionally, in order to take advantage of the categorisation mechanism proposed, we would need to compute all witnesses for each Σ-concept (which is relatively cheap in SubDiff).

7 Conclusions

We have formulated the problem of finding the set of affected terms between ontologies via model inseparability, and presented feasible approximations to finding this set. We have shown that each of the approximations can find considerably

more changes than those visible in a comparison of concept hierarchies. Both sound approximations devised capture more changes than Approx-CEX. The restrictions imposed by CEX on the input ontologies make change-preserving approximations a challenge, as we have seen in our attempt to reduce the NCIt to $\mathcal{EL}$ in a less naive way.

The proposed distinction between (purely) direct and indirect allows users to focus on those changes which are specific to a given concept, in addition to masking possibly uninteresting changes to any and all concept names (such as those obtained via witnesses constructed with negation and disjunction), thereby making change analysis more straightforward. As demonstrated by the NCIt change log analysis, we have found a (often high) number of direct changes that are not contained in the NCIt change logs, which leads us to believe the recording of changes does not seem to follow from even a basic concept hierarchy comparison, but rather a seemingly ad hoc mechanism.

In future work we aim to optimise the devised approximations so as to compare all NCIt versions w.r.t. their signature union, and deploy an end-user tool.

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. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. of Artificial Intelligence Research 31 (2008)
3. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. of KR-06 (2006)
4. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: Proc. of CIKM-11 (2011)
5. Horridge, M., Bechhofer, S.: The OWL API: A Java API for working with OWL 2 ontologies. In: Proc. of OWLED-09 (2009)
6. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga Llavori, R.: Supporting concurrent ontology development: Framework, algorithms and tool. Data and Knowledge Engineering 70(1) (2011)
7. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: IJCAR-08. vol. 5195 (2008)
8. Křemen, P., Šmíd, M., Kouba, Z.: OWLDiff: A practical tool for comparison and merge of OWL ontologies. In: Proc. of DEXA-12 (2011)
9. Küsters, R.: Non-Standard Inferences in Description Logics, LNAI, vol. 2100. Springer-Verlag (2001)
10. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: Proc. of IJCAI-07 (2007)
11. Lutz, C., Wolter, F.: Conservative extensions in the lightweight description logic $\mathcal{EL}$. In: Proc. of CADE-21 (2007)
12. Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., Parkinson, H.E.: Modeling sample variables with an experimental factor ontology. Bioinformatics 26(8), 1112–1118 (2010)
13. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness Preserving Approximation for TBox Reasoning. In: Proc. of AAAI-10 (2010)
14. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Proc. of DL-09 (2009)