Upwardly Abstracted Definition-Based Subontologies*

Ghadah Alghamdi[1][0000−0003−3836−1824],
Renate A. Schmidt[1][0000−0002−6673−3333], Warren Del-Pinto[1], and
Yongsheng Gao[2][0000−0002−3468−2930]

[1] The University of Manchester, UK
[2] IHTSDO (SNOMED International), UK

Abstract. In this paper, we present a method for extracting subontologies from $\mathcal{ELH}$ ontologies for a set of symbols. The approach is focused on the generation of upwardly abstracted definitions, which is a technique for computing definitions expressed using closest primitive ancestors. The subontologies returned by the method are evaluated for quality and compared to extracts computed with locality-based modularisation and uniform interpolation methods. Our subontology generation method produces promising results in terms of size and relevance to the needs of domain experts.

1 Introduction

Ontologies are formalised representations of domain knowledge. SNOMED CT, the Gene ontology (GO) and the NCIt ontology are just a few of the major ontologies used in the biomedical domain [7, 9, 12, 28, 30, 35]. Due to the large size and complexity of such ontologies, it is necessary to facilitate their use for a variety of applications including analysis, curation, debugging, and integration. To overcome the size issue, domain-specific subsets of concept names (reference sets) such as the ERA reference set [32] are used in SNOMED CT. Reference sets assist in limiting querying, searching, and data entry to a part of the application domain, and are carefully created to represent specific definitions from the source ontology indicating their intended function. Retrieving information for such reference sets requires querying the ontology in its entirety. In order to minimise the computational cost and overhead, rather than using a flat list of concepts, it is advantageous to be able to use instead a subontology that encompasses all semantic relationships associated with the concepts in the reference set.

There are various methods and approaches for extracting subontologies in the literature, including graph-based ontology partitioning [8], locality-based modularisation [10], and uniform interpolation [16]. Syntactic locality-based modularisation (SLBM) is a method that is widely used for the purpose of importability

---

We thank Yizheng Zhao and Zhao Liu for allowing us to use their $\mathcal{ELH}$ forgetting tool, and their help in using it.

and reuse. The method computes a subset of the stated axioms of an ontology that covers information related to the input signature. Modules can be large and can contain information outside the meaning of the input signature [10].

Another method for producing compact representations is uniform interpolation (UI), a logic-based method for computing restricted views of an ontology which faithfully captures information about the set of specified concepts and role names. This is accomplished by forgetting symbols that are not contained in the input signature [16,22,24]. Because the axioms within the UIs are rewritten during the forgetting process, the syntactic form of axioms may differ significantly from those in the original ontology.

Locality-based modularisation and uniform interpolation are useful for a variety of applications such as ontology summarisation, reuse, analysis, logical difference, and information hiding [6, 15, 18, 23, 34]. To be useful to the SNOMED community, subontologies must be in the language of SNOMED CT and must also satisfy SNOMED modelling guidelines. While modules satisfy these requirements because they contain only axioms from the original ontology, they tend to contain an excessive number of symbols that are not part of the input signature [4, 17, 21, 33]. By contrast, UIs contain only those symbols specified in the input signature but rewrite axioms and are harder to compute even when combined with modularisation [4]. To be beneficial to SNOMED users, a different notion of subontology is needed.

In this paper, we introduce a notion of subontology based on the idea of abstracted definitions, because SNOMED CT users are already familiar with a variety of normal forms [11, 27]. Our abstracted definitions follow the same format as the commonly used proximal primitive normal form in SNOMED CT. Proximal primitive normal forms explicitly state all possible constraints and defining characteristics of particular concepts to facilitate implementation, recording, storage, and retrieval within SNOMED CT. Additionally, such forms enable more precise inferred parent identification of focus concepts after running a classifier, they simplify parent relationship maintenance, and improve the accuracy and breadth of super and subconcepts [31]. We found that the generation of abstracted definitions aides the inclusion of information that is truly necessary for the resulting subontologies. Our method is targeted at acyclic ontologies such as SNOMED CT, the Gene ontology, and the Sequence ontology.

The main contributions of the paper are threefold:

- An investigation of computing upwardly abstracted definitions for $\mathcal{ELH}$ ontologies that satisfy common modelling guidelines used in SNOMED CT, one of the most widely used biomedical ontologies.
- A method for extracting subontologies based on the principle of abstracted definitions.
- An evaluation of the subontology extraction method and comparison with two existing subontology extraction methods, namely the bottom variant of SLBM, and the UI method, in order to comprehend benefits of subontologies based on the abstracted definitions.

The paper is organised as follows. Section 2 gives preliminary definitions and background information. Section 3 goes into detail about the upwardly abstracted definitions. The aim, requirements, algorithm and an example of the subontology generation method are presented in Section 4. Section 5 discusses related notions including SLBM and uniform interpolation. In Section 6, we evaluate the quality of the returned subontologies, and compare the results of our subontology generation method with SLBM and uniform interpolation in terms of precision and extract size. Finally, we conclude in Section 7.

2 Preliminaries and Background

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be disjoint sets of concept and role names respectively. The union of such sets form the signature of an ontology $\mathcal{O}$. The signature $\text{sig}(\xi)$ is a set of concept and role names that occur in ξ, where ξ is any syntactic object or ontology. The set of *$\mathcal{EL}$-concepts* C, and the sets of *$\mathcal{ELH}$-axioms* α are built according to the grammar rules: $C ::= A \mid C \sqcap C \mid \exists r.C$ and $\alpha ::= C \sqsubseteq C \mid C \equiv C \mid r \sqsubseteq s$ where $A \in \mathsf{N_C}$ and $r, s \in \mathsf{N_R}$.[1] An *$\mathcal{ELH}$-TBox* is a finite set of $\mathcal{ELH}$-axioms. The semantics, including the notions of model, satisfaction of concepts, axioms and TBoxes as well as the logical consequence relation (entailment), are defined in the usual manner; see, for example [2].

A *terminology* is a TBox that contains only axioms of the form $A \equiv C$ or $A \sqsubseteq C$, with A appearing not more than once on the left-hand side of an axiom. If A does not depend on itself, i.e., does not occur in the set of symbols required to define itself for any $A \in \mathsf{N_C}$, the terminology is *acyclic*.

Classification is a standard reasoning task that computes a hierarchy $\mathcal{H}$ for $\mathcal{O}$. A hierarchy $\mathcal{H}$ is a finite set of subsumption axioms $A \sqsubseteq B$ such that $\mathcal{O} \models A \sqsubseteq B$, where A and B are concept names in $\text{sig}(\mathcal{O})$.

Figure 1 shows an example of an $\mathcal{ELH}$ biomedical terminology, adapted from [5]. The axioms α_1 to α_5 are concept definitions of the form $A \equiv C$ or $A \sqsubseteq C$, where A is the described concept. Definitions, with both necessary and sufficient conditions ($A \equiv C$), are critical in terminologies because they assist in determining which concepts are classified under them in the concept hierarchy. On the other hand, concepts with necessary conditions only ($A \sqsubseteq C$) affect how a concept is classified, but have no effect on which concepts can be classified under them [25].

Our study on generating *upwardly* abstracted definitions is limited to axioms of the form $A \equiv C$ or $A \sqsubseteq C$ in an *$\mathcal{ELH}$ ontology* or *terminology*, as the majority of biomedical ontologies lack GCIs of the form $C \sqsubseteq A$, $C \sqsubseteq D$, or $C \equiv D$, where C and D denote complex concepts and A denotes a concept name.

α_1 : **InflammatoryDisorder** $\equiv$ Disease $\sqcap$ $\exists$involves.Inflammation,
α_2 : **LiverDisease** $\equiv$ Disease $\sqcap$ $\exists$location.Liver,
α_3 : **Hepatitis2** $\equiv$ LiverDisease $\sqcap$ $\exists$involves.Inflammation,
α_4 : **LargeLiver** $\sqsubseteq$ LiverDisease $\sqcap$ $\exists$location.EntireLiver,
α_5 : **EntireLiver** $\sqsubseteq$ Liver

Fig. 1. Fragment of a biomedical terminology illustrating the $\mathcal{ELH}$ logical constructs

---

[1] These grammar rules are sufficient for the $\mathcal{ELH}$ fragment we are considering.

3 Upwardly Abstracted Definitions

We start by defining the key concepts used in this article.

Definition 1 (Defined (Primitive) Concept). *Let $\mathcal{O}$ be an ontology, and C an $\mathcal{EL}$-concept other than A. A concept name A is a* defined concept *in $\mathcal{O}$ if there is an axiom of the form $A \equiv C$ in $\mathcal{O}$. Otherwise, it is called a* primitive concept*.*

Regardless as to whether a named concept A is defined or primitive, its abstracted definition is computed by inferring its closest primitive ancestor(s) in the subsumption hierarchy. Closest primitive ancestors are defined as follows.

Definition 2 (Closest Primitive Ancestor). *Let $\mathcal{O}$ be an ontology, $A, P \in sig(\mathcal{O})$ where A is a defined or a primitive concept name and P is a primitive concept name. We say that P is a* closest primitive ancestor *to A in $\mathcal{O}$ if $\mathcal{O} \models A \sqsubseteq P$ and there does not exist a primitive concept name $Z \in sig(\mathcal{O})$ (other than P or A) such that $\mathcal{O} \models A \sqsubseteq Z$ and $\mathcal{O} \models Z \sqsubseteq P$. The set of closest primitive ancestors to A will be denoted by $\mathcal{P}_A$.*

We define abstracted definitions as follows.

Definition 3 (Upwardly Abstracted Definition). *Let $\mathcal{O}$ be an ontology, and A a defined (primitive) concept name in $\mathcal{O}$. The* abstracted definition *of A is $A \equiv \mathcal{P}_A \sqcap \mathcal{E}_A$ ($A \sqsubseteq \mathcal{P}_A \sqcap \mathcal{E}_A$), where $\mathcal{P}_A$ is a conjunction of the closest primitive ancestors to A, while $\mathcal{E}_A$ is a conjunction of existential restrictions (of the form $\exists r.C$) required to complete the abstracted definition of A such that $\mathcal{O} \models A \equiv \mathcal{P}_A \sqcap \mathcal{E}_A$ (or $\mathcal{O} \models A \sqsubseteq \mathcal{P}_A \sqcap \mathcal{E}_A$), and $sig(\mathcal{P}_A \sqcap \mathcal{E}_A) \subseteq sig(\mathcal{O})$.*

Example 1. Consider the ontology $\mathcal{O} = \{A \equiv D \sqcap \exists r.C_1, D \equiv P \sqcap \exists r.C_2, P \sqsubseteq \exists r.C_3\}$. An upwardly abstracted definition of A is $A \equiv P \sqcap \exists r.C_1 \sqcap \exists r.C_2 \sqcap \exists r.C_3$.

Different equivalent logical forms of SNOMED CT concept definitions are discussed in [27]. Two distinct forms of proximal primitive modelling is mentioned there: first, a *short canonical form* in which only the existential restrictions that distinguish the concept from its closest primitive ancestors are listed. For instance, in Example 1, the short canonical form of A is $A \equiv P \sqcap \exists r.C_1 \sqcap \exists r.C_2$. The second is the *long canonical form*, which lists all possible existential restrictions, which can be viewed as the defining characteristics of the concept being defined [27]. In Example 1, the long canonical form of A is $A \equiv P \sqcap \exists r.C_1 \sqcap \exists r.C_2 \sqcap \exists r.C_3$. According to Definition 3, both short and long canonical forms are abstracted definitions. This illustrates that abstracted definitions are not unique.

The following example illustrates that abstracted definitions may also be weaker than the original definitions when $\mathcal{O}$ is an *ontology* rather than a *terminology.*

Example 2. Let $\mathcal{O} = \{\alpha_1, \alpha_2, \alpha_3\}$, where α_1: $A \equiv D \sqcap \exists r.C$, α_2: $D \equiv P \sqcap \exists r.C$ and α_3: $A \sqsubseteq P_2$. We notice that both ab_1: $A \equiv P \sqcap \exists r.C$ and ab_2: $A \equiv P \sqcap P_2 \sqcap \exists r.C$ are entailed by $\mathcal{O}$ and are in fact abstracted definitions of A. Let's consider the ontologies $\mathcal{O}_1 = \{ab_1, \alpha_2, \alpha_3\}$ and $\mathcal{O}_2 = \{ab_2, \alpha_2, \alpha_3\}$ in which the original definition α_1 of A is respectively replaced by ab_1 and ab_2. We observe that $\mathcal{O} \equiv \mathcal{O}_1$ but $\mathcal{O} \not\equiv \mathcal{O}_2$ because $\mathcal{O}_2 \not\models \mathcal{O}$ as $\mathcal{O}_2 \not\models \alpha_1$ since ab_2 is weaker than α_1.

The abstracted definition of A in $\mathcal{O}$ computed by our algorithm is the second case (ab_2) as part of its search for all of the closest primitive ancestors. As seen in the example, this definition is weaker than the original definition. To define the logical strength of abstracted definitions, we use the following definition:

Definition 4 (Logical Strength). *An ontology $\mathcal{O}'$ is* weaker *than another ontology $\mathcal{O}$ if $\mathcal{O} \models \mathcal{O}'$ but $\mathcal{O}' \not\models \mathcal{O}$. An axiom α' is* weaker *than another axiom α in $\mathcal{O}$ if $\mathcal{O} \models \alpha'$ but $\mathcal{O} \backslash \{\alpha\} \cup \{\alpha'\} \not\models \alpha$.*

4 Computing Subontologies

Our aim is to compute for a given set of symbols a domain-specific *subontology* from a source ontology that satisfies the following requirements:

1. The subontology must capture the meaning of the concepts in the focus set, using whenever possible abstracted definitions in long canonical form.
2. The transitive closure of concept name subsumption of the subontology is a restriction of the original ontology's transitive closure of concept name subsumption over the signature of the subontology.

These requirements were established with a leading terminologist at SNOMED international.

Our method to compute subontologies is presented in Algorithm 1. The algorithm takes as input an ontology $\mathcal{O}$ and a focus set Σ_F of concept and role names to generate a subontology $\mathcal{S}$. The first step of the algorithm initialises the output $\mathcal{S}$, and the set of existential restrictions $\mathcal{E}$, and Σ^+ is set to Σ_F. The second step classifies $\mathcal{O}$ using the ELK reasoner [14] to obtain the concept hierarchy $\mathcal{H}$, which is then used throughout the algorithm to compute the abstracted definitions for focus concepts correctly, i.e., to compute concepts that can be related to a focus concept via inferred relations. As a result, the correct subontology subsumption hierarchy is derived.

Computing an abstracted definition for a focus concept $A \in \Sigma_F$ calls the method AbstractedDefinitionExtraction presented in Algorithm 2. The method starts with computing the Ancestors of A using $\mathcal{H}$. The Ancestors set consists of all concept names in $\mathrm{sig}(\mathcal{O})$ that subsume A. In Line 2, the function ComputePrimitiveAncestors filters the set of ancestors by determining their status in $\mathcal{O}$ as defined or primitive in order to obtain just the primitive concept ancestors. We use $\mathcal{O}$ to obtain the existential restrictions $\mathcal{E}_A$ that occur in the right-hand side of A's definition and all of A's ancestors' definitions, which is computed by the function ComputeExistentialRestrictions in Line 3.

Algorithm 1 SubontologyExtraction($\mathcal{O}$, Σ_F)

```
Input: Ontology $\mathcal{O}$, Focus set $\Sigma_F$
Output: Subontology $\mathcal{S}$
1: $\mathcal{S} := \emptyset$, $\Sigma^+ := \Sigma_F$, $\mathcal{E} := \emptyset$.
2: $\mathcal{H} := \mathsf{Classify}(\mathcal{O})$
3: for $A \in \Sigma_F$ do
4:     if $A$ is a defined concept name in $\mathcal{O}$ then
5:         $A \equiv \mathcal{P}'_A \sqcap \mathcal{E}'_A := \mathsf{AbstractedDefinitionExtraction}(A, \mathcal{O}, \mathcal{H}, \mathsf{TRUE})$
6:     else $A \sqsubseteq \mathcal{P}'_A \sqcap \mathcal{E}'_A := \mathsf{AbstractedDefinitionExtraction}(A, \mathcal{O}, \mathcal{H}, \mathsf{FALSE})$
7:     $\Sigma^+ := \Sigma^+ \cup \mathsf{sig}(A \equiv (\sqsubseteq)\ \mathcal{P}'_A \sqcap \mathcal{E}'_A)$
8:     $\mathcal{E} := \mathcal{E} \cup \mathsf{GetExistentialRestrictions}(A \equiv (\sqsubseteq)\ \mathcal{P}'_A \sqcap \mathcal{E}'_A)$
9:     $\mathcal{S} := \mathcal{S} \cup A \equiv (\sqsubseteq)\ \mathcal{P}'_A \sqcap \mathcal{E}'_A$
10: $\mathcal{S} := \mathcal{S} \cup \mathsf{ComputeAdditionalAxioms}(\Sigma^+, \mathcal{E}, \mathcal{H}, \mathcal{O})$
```

Algorithm 2 AbstractedDefinitionExtraction(A, $\mathcal{O}$, $\mathcal{H}$, *isDefinedConcept*)

```
Input: The concept to define $A$, Ontology $\mathcal{O}$, Concept hierarchy $\mathcal{H}$, Defined concept checker isDefinedConcept
Output: The abstracted definition of $A$
1: $\mathsf{Ancestors} := \mathsf{ComputeAncestorsOf}A(A, \mathcal{H})$
2: $\mathcal{P}_A := \mathsf{ComputePrimitiveAncestors}(\mathsf{Ancestors}, \mathcal{O})$
3: $\mathcal{E}_A := \mathsf{ComputeExistentialRestrictions}(A, \mathsf{Ancestors}, \mathcal{O})$
4: $\mathcal{P}'_A \sqcap \mathcal{E}'_A := \mathsf{RemoveRedundantConcepts}(\mathcal{P}_A, \mathcal{E}_A, \mathcal{H}, \mathcal{O})$
5: if isDefinedConcept then
6:     return $A \equiv \mathcal{P}'_A \sqcap \mathcal{E}'_A$
7: else return $A \sqsubseteq \mathcal{P}'_A \sqcap \mathcal{E}'_A$
```

The method RemoveRedundantConcepts in Line 4 returns the set $(\mathcal{P}'_A \sqcap \mathcal{E}'_A)$, which is a conjunction of the closest primitive ancestors and the most specific existential restrictions, after the removal of possible redundant concepts from the set of primitive concepts $\mathcal{P}_A$ and existential restrictions $\mathcal{E}_A$. A concept D is regarded as redundant if it occurs in another concept C where D is equivalent to or subsumes C. Redundant concepts in the sets $\mathcal{P}_A$ and $\mathcal{E}_A$ are removed according to the general rule $C \sqcap D \equiv C \Leftrightarrow C \sqsubseteq D$. The removal of redundant concepts from the set $\mathcal{P}_A$ results in the set of closest primitive ancestors. To remove redundant existential restrictions from the set $\mathcal{E}_A$, we follow the rules in Figure 2. For example, if the set $\mathcal{E}$ has two existential restrictions, $E := \exists t.(\exists r_1.A_1 \sqcap \ldots \sqcap \exists r_n.A_n)$ and $G := \exists u.(\exists s_1.B_1 \sqcap \ldots \sqcap \exists s_m.B_m)$, we check whether $E \sqsubseteq G$. We do this by using the first rule to check if the outer role t in E is equivalent to or subsumed by the outer role u in G. If this is the case, then we continue with the first rule to check if an existential restriction $\exists r_i.A_i$ is subsumed by or equivalent to an existential restriction $\exists s_j.B_j$ under the nested roles t and u, respectively. Then, following rule 2, we determine if the subsumption checking performed using rule 1 is sufficient for all $\exists r_i.A_i$ to be subsumed by all $\exists s_j.B_j$.

If this is the case, then it means that $E \sqsubseteq G$, and G can be removed from the set $\mathcal{E}$.

$$1.r \sqsubseteq s, C \sqsubseteq D \Rightarrow \exists r.C \sqsubseteq \exists s.D$$
$$2.\forall i = 1..n\ \exists j = 1..m\ \exists r_i.C_i \sqsubseteq \exists s_j.D_j \Rightarrow \exists t.(\exists r_1.C_1 \sqcap .. \sqcap \exists r_n.C_n) \sqsubseteq \exists t.(\exists s_1.D_1 \sqcap .. \sqcap \exists s_m.D_m)$$

Fig. 2. Properties of subsumption used as rules to remove possible redundant existential restrictions where C and D are $\mathcal{EL}$-concepts and r, s and t $\in \mathsf{N_R}$

Algorithm 2 concludes by returning the abstracted definition of A as either $A \equiv \mathcal{P}'_A \sqcap \mathcal{E}'_A$ or $A \sqsubseteq \mathcal{P}'_A \sqcap \mathcal{E}'_A$ depending on whether A is a defined or a primitive concept in $\mathcal{O}$. Lines 7 and 8 of Algorithm 1 add the signature, and the existential restrictions of the generated abstracted definition to the sets Σ^+ and $\mathcal{E}$, respectively. Line 10 returns the subontology $\mathcal{S}$ after using the function ComputeAdditionalAxioms to add extra axioms to complete the hierarchy of the subontology. This is performed by looking for possible subsumption relations between concept and role names in the set Σ^+, as well as between concept names in Σ^+ and the existential restrictions in $\mathcal{E}$. For example, the axiom α_3 in Figure 3 is an additional axiom that is required for the subontology hierarchy to be complete.

Our method abstracts the definitions based on the closest primitive ancestors in order to include only what is truly necessary in the resulting subontology. For example, if a user is interested in computing a subontology using the ontology in Figure 1 for the focus set concepts A_1:**Hepatitis2** and A_2:**LargeLiver**, then applying Algorithm 1 for A_1 and A_2 results in the subontology shown in Figure 3. It includes abstracted definitions of A_1 and A_2, where the concept A_3:LiverDisease occurring in the original definitions of A_1 and A_2 has been abstracted away. The use of A_3 would require the inclusion of its definition and increase the size of the desired subontology without adding additional meaning. This is because A_3 carries the same information that A_1 and A_2 inherit, and thus, the definition of A_3 is superfluous with respect to the *focus set* $\{A_1, A_2\}$ of interest to the user. We refer to the extra symbols occurring in the signature of the generated focus set definitions as the *supporting set*, e.g., the concept *EntireLiver* occurring in the definition of **LargeLiver** is a supporting concept.

α_1 : **Hepatitis2** $\equiv$ Disease $\sqcap$ $\exists$involves.Inflammation $\sqcap$ $\exists$location.Liver
α_2 : **LargeLiver** $\sqsubseteq$ Disease $\sqcap$ $\exists$location.EntireLiver
α_3 : EntireLiver $\sqsubseteq$ Liver

Fig. 3. The subontology for focus concepts A_1, A_2

We define a subontology for a given set of focus symbols Σ_F as follows.

Definition 5 (Focus Set Subontology). *Let $\mathcal{O}$ be an $\mathcal{ELH}$ ontology and Σ_F a focus set. $\mathcal{S}$ is a* focus set subontology *of $\mathcal{O}$ w.r.t. Σ_F if the following conditions are satisfied: (i) $\Sigma_F \subseteq sig(\mathcal{S})$; (ii) for every $\mathcal{ELH}$-axiom α where $sig(\alpha) \subseteq sig(\mathcal{S})$ we have: (a) $\mathcal{S} \models \alpha \implies \mathcal{O} \models \alpha$ and (b) if α is of the form $A \sqsubseteq B$, then $\mathcal{O} \models \alpha \implies \mathcal{S} \models \alpha$, where A and B are concept names.*

Depending on the input ontology, our algorithm generates two types of subontologies: *focus set subontologies*, which may contain weaker abstracted definitions (cf. ab_2 in $\mathcal{O}_2$ in Example 2), and *equivalent focus set subontologies*, which contain abstracted definitions that are equivalent to their original definitions in $\mathcal{O}$ for all concepts in the focus set Σ_F. We define a subontology that contain equivalent focus set abstracted definitions for all focus concepts in Σ_F as follows.

Definition 6 (Equivalent Focus Set Subontology). *Let $\mathcal{O}$ be an $\mathcal{ELH}$-acyclic terminology, and $\mathcal{S}$ a focus set subontology of $\mathcal{O}$ w.r.t. Σ_F. We say that $\mathcal{S}$ is an* equivalent focus set subontology *when $\mathcal{O} \models \alpha \Leftrightarrow \mathcal{S} \models \alpha$ for all abstracted definitions α of focus concepts in Σ_F in $\mathcal{S}$.*

In order to obtain an equivalent focus set subontology, the input ontology to our method has to be $\mathcal{ELH}$-acyclic *terminology.*

5 Related Notions

5.1 Syntactic Locality Based Modularisation

SLBM is widely used in ontology engineering to facilitate ontology reuse and import [26]. It is available as part of the OWL API tool [13]. The method takes as input an ontology $\mathcal{O}$ and a seed signature Σ, and supports computing three distinct module types: bottom ($\bot$), top ($\top$), and nested ($\bot\top*$) modules. In essence, SLBM internally extends Σ to cover the upward (downward) views of Σ until it reaches the top (bottom) symbol in $\mathcal{O}$, as specified by the $\bot$ ($\top$) types. The $\bot\top*$-type is an iteration of $\top$ and $\bot$ types, returning a module with symbols contained within Σ's $\bot$ ($\top$) views. We use the $\bot$ type in our evaluation (Section 6) because it is the most relevant type for how our method works, which is to generate an upwardly expanded extract for an input focus set.

SLBM retains the original forms of axioms in the resulting modules. However, for the purpose of extracting definitions of concepts in Σ, modules computed contain a significant number of supporting symbols. For instance, as demonstrated in [4], computing $\bot\top*$-modules from SNOMED CT using NHS subsets results in modules with a precision rate of 72%. Very low average precision rate (1.14%) was obtained in a research examining module extraction using a medical corpus annotated with SNOMED CT concepts, performed for usability purposes [21].

5.2 Uniform Interpolation

Uniform interpolation is a task that allows potentially undesirable symbols to be eliminated from an ontology without affecting the meaning of the remaining symbols in the ontology [16,19,22,24,36]. Theoretically, it has been proved that the size of the given UIs can be exponentially three times larger than the size of the input ontology [24]. However, an evaluation in [3] with real-world signatures demonstrated that the size is less than the original ontology, with a precision rate of 100% for all resulting UIs except for a few that contained definer names [4].

Computing UIs for focus set concepts produces ontologies too small to include definitions of the focus concepts. For example, computing a UI for A_1: *Hepatitis2* and A_2:*LargeLiver* for the ontology in Figure 1, requires forgetting the rest of the ontology's symbols. This results in the UI $\{A_1 \sqsubseteq \top, A_2 \sqsubseteq \top\}$, because the RHSs of the definitions of A_1 and A_2 are eliminated. More informative UIs can be generated by first extending the focus set using a signature extension algorithm as in [4]. However, the resulting UIs had rewritten axioms (not in their original form), which were not satisfactory for SNOMED CT users.

6 Evaluation

The goal of the evaluation is to test empirically the following three hypotheses about our method: (1) The abstracted definitions in the generated subontologies are equivalent to their original definitions when the input ontology is a terminology (Section 6.1). (2) Abstracted definitions aid in the reduction of axioms deemed redundant in relation to the input focus set. The study of the signature of the bottom modules demonstrates this (Section 6.2). (3) Subontologies are smaller in size than bottom modules, and smaller or equal in size to the UIs (Sections 6.2 and 6.3). To do this, we developed a Java prototype implementing our method using the OWL API. [2] We employed two distinct ontologies, SNOMED CT (July 2017) and the Gene ontology (GO) (February 2021). GO does not meet the requirements of a terminology, as it may contain more than one axiom for a concept name A. Both SNOMED CT and GO lack GCI axioms of the forms $C \sqsubseteq A$ and $C \sqsubseteq D$ where C and D are $\mathcal{EL}$-concepts.

We compared the produced subontologies to the two extract types outlined in Section 5: the $\bot$-modules and the UIs. We computed UIs using a newly developed UI tool for $\mathcal{ELH}$ ontologies [20]. The $\bot$-modules were generated using the OWL API's built-in SLBM tool. The comparisons involve a size and precision rate analysis of the generated extracts.

SNOMED CT had a total of 335 245 logical axioms, 335 225 concept names and 97 role names. The Gene ontology had a total of 102 203 logical axioms, 44 085 concept names,[3] and 8 role names. Both ontologies are in the scope of $\mathcal{ELH}$, with the exception of one role chain axiom ($r \circ s \sqsubseteq r$) in SNOMED CT and one inverse role, 29 disjoint classes, four transitive roles, and two role chain axioms in the Gene ontology. Non $\mathcal{ELH}$-axioms were omitted.

As focus sets, we used five sets of human and animal medical conditions for computing SNOMED CT extracts used in the experiments of [1]. The supplementary information of these experiments includes a list of 20–40 concept names for each of the medical conditions.[4] Due to the small number of concept names provided, we computed descendant concepts of these to increase the size of the generated extracts, which can provide better insights into the types of extracts we considered. As a result, the focus sets used consisted of 4401–14 828 concept

---

[2] http://owlapi.sourceforge.net/

[3] This number excludes 6 430 deprecated concepts that exist in the version 01-02-2021

[4] `https://tinyurl.com/medical-conditions-signature`

Table 1. Results of logical strength test of the abstracted definitions of medical conditions and gene slim focus sets in Snomed CT (SCT) and Gene Ontology (GO), respectively.

Quality criteria	Σ_F definitions' logical strength		
Focus set	# Equivalent Σ_F definitions	# Weaker Σ_F definitions	Total Σ_F
Anaemia (SCT)	4401	0	4401
Arthritis (SCT)	7805	0	7805
Diabetes (SCT)	74291	0	74291
Hypertension (SCT)	14828	0	14828
Obesity (SCT)	9885	0	9885
goslim_mouse (GO)	42	2	44
goslim_pir (GO)	452	1	453
goslim_plant (GO)	96	0	96
goslim_pombe (GO)	56	2	58
goslim_yeast (GO)	160	3	163

Table 2. Number of logical axioms, concepts and roles in the different types of extracts for five focus sets in Snomed CT and Gene Ontology.

Entity type	Logical Axioms	Concepts	Roles	Logical Axioms	Concepts	Roles	Logical Axioms	Concepts	Roles
SNOMED CT	Subontologies			Bottom modules			UIs		
Min.	6 712	6 398	19	17 706	17 699	38	N/A	N/A	N/A
Max.	86 819	86 608	46	102 628	102 617	55	N/A	N/A	N/A
Avg.	27 242.4	27 149	31.4	39 219	39 210	47	N/A	N/A	N/A
Med.	13 816	13 904	31	22 917	22 907	47	N/A	N/A	N/A
GENE ONTOLOGY	Subontologies			Bottom modules			UIs		
Min.	79	82	2	467	301	8	82	82	2
Max.	639	641	4	2 243	1 554	8	746	641	4
Avg.	259.4	262	3.2	12 37.8	809.2	8	302.8	262	3.2
Med.	155	158	3	1 021	626	8	172	158	3

names. For the Gene ontology, we used five focus sets of the GO slim sets provided in [29]. Each set represents a flat list of concept and role names that are specific to certain species or organisms. All experimental data used in this study is available at `https://tinyurl.com/evaluation-data`.

Given how the three methods work, we used the following input sets to generate the different extracts: (1) The focus set was used as the input signature for computing the bottom modules and subontologies, because both SLBM and our subontology generation method extend the signature as needed. (2) As input to the UI method, we used the signatures of the computed subontologies. As mentioned, UIs for focus sets would not adequately capture their definitions.

6.1 Logical Strength of the Upwardly Abstracted Definitions

To determine the logical strength of the generated abstracted definitions, the test of Definition 4 was used. We found that when using SNOMED CT to compute subontologies, the abstracted definitions generated have the same logical strength as the original definitions. This is because SNOMED CT is a terminology that includes no more than one concept definition for a concept name A. On the other hand, when the Gene ontology is used, the abstracted definitions derived may be weaker than the original definitions. The count numbers of the logical strength of the abstracted definitions in the subontologies of SNOMED CT and the Gene ontology are shown in Table 1.

Table 3. The values of Precision, |Region|, $|\text{sig}(\mathcal{S})|$ and $|\text{sig}(\mathcal{M})|$ of different medical conditions and gene slim focus sets in SNOMED CT (SCT) and GENE ONTOLOGY (GO), respectively.

Focus set	Precision	|Region|	$\|\text{sig}(\mathcal{S})\|$	$\|\text{sig}(\mathcal{M})\|$
Anaemia (SCT)	0.28	12%, 2006	6436	22960
Arthritis (SCT)	0.51	22%, 1887	9199	17737
Diabetes (SCT)	0.84	99.9%, 16030	86627	102664
Hypertension (SCT)	0.58	14%, 2025	19705	33758
Obesity (SCT)	0.74	22%, 1157	13935	19166
goslim_mouse (GO)	0.26	1%, 3	84	309
goslim_pir (GO)	0.41	2%, 20	645	1562
goslim_plant (GO)	0.31	3%, 9	161	501
goslim_pombe (GO)	0.22	1%, 3	145	634
goslim_yeast (GO)	0.26	2%, 16	291	1080

6.2 Subontology Results Against Bottom SLBM

Table 2 shows that the number of symbols (concepts and roles) in the bottom modules were significantly larger than those in the subontologies. The average numbers of logical axioms, concepts, and roles in SNOMED CT's subontologies were 36.04%, 36.35%, and 39.79% less than the average numbers of those in the bottom modules, respectively.

Table 3 shows the computed values for Precision and Region for each bottom module, as well as the number of signatures for subontologies denoted by $|\text{sig}(\mathcal{S})|$ and for bottom modules denoted by $|\text{sig}(\mathcal{M})|$ for each SNOMED CT and Gene ontology focus set. We use the formula given in [4] to define Precision(extract, $\mathcal{S}$) as $(|(\text{sig}(\mathcal{S}) \cap \text{sig}(\text{extract}))| \cup \{\top\}/|\text{sig}(\text{extract})|)$. This gives the ratio of relevant symbols that occur in the signature of a subontology over the number of symbols in an extract. We regard an extract to be precise if it contains no symbols that are not part of the subontology's signature. This is because subontologies' signatures correspond to the focus set definitions and supporting set inclusions, which represent the necessary information about the input focus set that a user is interested in. As can be seen from Table 3, precision ratings for the bottom modules range between 28% and 84% for SNOMED CT focus sets and are almost half that for the Gene ontology focus sets (22%–41%).

To quantify the benefits of abstracting focus concepts' definitions in terms of reducing the size of an extract, we examined the set of symbols in the bottom modules that occur outside the set of their corresponding subontologies' signatures. Specifically, we calculated the number of concepts that exist in the region between the concepts in the focus set Σ_F and their closest primitive ancestors, denoted by $\mathcal{P}_{\Sigma_F}$ using the formula:

$$\text{Region}(\Sigma_F, \mathcal{P}_{\Sigma_F}) := (\text{Ancestors of } \Sigma_F \cap \text{Descendants of } \mathcal{P}_{\Sigma_F}).$$

The definitions of concepts found in the Region set can be viewed as redundant information in relation to the input focus set definitions. This is because definitions of concepts in the Region set should carry the same information that the abstracted definitions of focus concepts inherit, which is especially true when the input ontology is a terminology. Thus, incorporating definitions of focus concepts following abstraction is sufficient for including information that is genuinely important in the extract, as the abstraction process aids in the inclusion of all of the focus concept's defining characteristics. As can be observed from Table 3, 99.9% of concepts that exist outside the subontology's signature in the *Diabetes* bottom module occur in the Region set, demonstrating that the size of such an extract can be greatly reduced by abstracting the definitions of focus concepts to their closest primitive ancestors.

6.3 Subontology Results Against UI

The UI tool was unable to generate views for the SNOMED CT medical conditions focus sets (Table 2). This is due to the fact that forgetting becomes more difficult when the input signature is small in comparison to the source ontology's signature, particularly when forgetting from a very complex, large ontology such as SNOMED CT [4].

As shown in Table 2, the number of concepts and roles for subontologies and UIs for the Gene ontology focus sets coincide, as the UIs were computed for the subontologies' signatures, confirming the expected 100% precision of UI. On the other hand, the results show that the number of logical axioms in UIs is greater than that in subontologies. There are two main explanations for this observation. The first is that the UI method has incorporated inferred axioms of the form $C \sqsubseteq A$, where A denotes a focus or a supporting concept name. For example, forgetting B_1 and B_2 from $B_1 \equiv C$, $B_1 \sqsubseteq B_2$, $B_2 \sqsubseteq A$, where all symbols in C are in the signature of the subontology, derives $C \sqsubseteq A$. The second is that when a focus concept A is described by multiple axioms, our method generated an abstracted definition that condenses these multiple axioms into a single axiom. For instance, the UI might contain two axioms for A, $A \equiv C_1$ and $A \sqsubseteq C_2$, while our method infers the axiom $A \equiv C_1 \sqcap C_2$, which results in fewer axioms in the subontology, but can also result in a weaker definition for A.

7 Conclusion

We presented a method of extracting subontologies, which is based on abstracted definitions for given sets of focus symbols. Such abstracted definitions are motivated by normal forms used in proximal primitive modelling in the SNOMED CT community. We empirically demonstrated that when the input ontology is a terminology, the method generates equivalent focus set subontologies; however, when the input ontology is not a terminology, the method may yield subontologies with weaker definitions.

In comparison to bottom modules, our abstracted definition-based subontologies contain significantly fewer supporting set symbols and contained fewer axioms. The sizes of the Region set in the bottom modules give additional insights into why the notion of abstracted definitions in subontologies produce significantly smaller extracts than bottom modules while retaining all of the defining characteristics of the focus set concepts.

References

1. Arguello-Casteleiro, M., Stevens, R., Des-Diz, J., Wroe, C., Fernandez-Prieto, M.J., Maroto, N., Maseda-Fernandez, D., Demetriou, G., Peters, S., Noble, P.J.M., Jones, P.H., Dukes-McEwan, J., Radford, A.D., Keane, J., Nenadic, G.: Exploring semantic deep learning for building reliable and reusable one health knowledge from pubmed systematic reviews and veterinary clinical notes. Journal of Biomedical Semantics (2019)
2. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
3. Chen, J., Alghamdi, G., Schmidt, R.A., Walther, D., Gao, Y.: Modularity Meets Forgetting: A Case Study with the SNOMED CT Ontology. In: Proceedings of the 32nd International Workshop on Description Logics. vol. 2373. CEUR-WS.org (2019)
4. Chen, J., Alghamdi, G., Schmidt, R.A., Walther, D., Gao, Y.: Ontology extraction for large ontologies via modularity and forgetting. In: Proceedings of the 10th International Conference on Knowledge Capture, K-CAP. pp. 45–52. ACM (2019)
5. Cornet, R., Abu-Hanna, A.: Auditing description-logic-based medical terminological systems by detecting equivalent concept definitions. International Journal of Medical Informatics **77 5**, 336–45 (2008)
6. Cuenca Grau, B., Motik, B.: Reasoning over Ontologies with Hidden Content: The Import-by-Query Approach. Journal of Artificial Intelligence Research **abs/1401.5853** (2014)
7. Eilbeck, K., Lewis, S., Mungall, C., Yandell, M., Stein, L., Durbin, R., Ashburner, M.: The Sequence Ontology: a tool for the unification of genome annotations. Genome biology **6**, R44 (2005)
8. Garcia, A.C., Tiveron, L., Justel, C., Cavalcanti, M.C.: Applying Graph Partitioning Techniques to Modularize Large Ontologies. In: ONTOBRAS-MOST. vol. 938, pp. 72–83. CEUR Workshop Proceedings (2012)
9. Golbeck, J., Fragoso, G., Hartel, F.W., Hendler, J.A., Oberthaler, J., Parsia, B.: The National Cancer Institute's Thésaurus and Ontology. J. Web Semant. **1**, 75–80 (2003)
10. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular Reuse of Ontologies: Theory and Practice. J. Artif. Intell. Res. **31**, 273–318 (2008)
11. Harry, M.: Primitive parent modelling. `https://confluence.ihtsdotools.org/display/DOCEG/Proximal+Primitive+Modeling`, SNOMED International, Accessed: 2021-05-23
12. Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P., Steinbeck, C.: ChEBI in 2016: Improved services and an expanding collection of metabolites. Nucleic Acids Research **44**, pp. D1214 – D1219 (2016)
13. Horridge, M., Bechhofer, S.: The OWL API: A Java API for Working with OWL 2 Ontologies. In: Proceedings of the 6th International Conference on OWL: Experiences and Directions. vol. 2, pp. 11–21 (2009)
14. Kazakov, Y., Krötzsch, M., Simancik, F.: The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with $\mathscr{EL}$ Ontologies. J. Autom. Reason. **53**, 1–61 (2014)
15. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The Logical Difference for the Lightweight Description Logic EL. Journal of Artificial Intelligence Research **44**, 633–708 (2012)

16. Konev, B., Walther, D., Wolter, F.: Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July (IJ-CAI). pp. 830–835 (2009)
17. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic Inseparability and Modularity of Description Logic Ontologies. Artif. Intell. **203**, 66–103 (2013)
18. Konev, B., Walther, D., Wolter, F.: The Logical Difference Problem for Description Logic Terminologies. In: Automated Reasoning. vol. 5195, pp. 259–274. Springer (2008)
19. Koopmann, P., Schmidt, R.A.: Forgetting Concept and Role Symbols in $\mathcal{ALCH}$-Ontologies. In: Proc. of LPAR'13. vol. 8312, pp. 552–567. Springer (2013)
20. Liu, Z., Lu, C., Alghamdi, G., Schmidt, R.A., Zhao, Y.: Tracking Semantic Evolutionary Changes in Large-Scale Ontological Knowledge Bases. To appear in the 30th ACM International Conference on Information and Knowledge Management (CIKM) (2021)
21. López-García, P., Boeker, M., Illarramendi, A., Schulz, S.: Usability-driven pruning of large ontologies: the case of SNOMED CT. Journal of the American Medical Informatics Association **19** (2012)
22. Ludwig, M., Konev, B.: Practical Uniform Interpolation and Forgetting for $\mathcal{ALC}$ TBoxes with Applications to Logical Difference. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria. AAAI Press (2014)
23. Lutz, C., Wolter, F.: Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain. pp. 989–995. IJCAI/AAAI (2011)
24. Nikitina, N., Rudolph, S.: (Non-)Succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. Artif. Intell. **215**, 120–140 (2014)
25. Rector, A.: Coordinating Taxonomies: Key to Re-Usable Concept Representations. In: Proceedings of Artificial Intelligence Medicine, 5th Conference on Artificial Intelligence in Medicine in Europe, AIME'95, Pavia, Italy. vol. 934, pp. 17–28. Springer (1995)
26. Sattler, U., Schneider, T., Zakharyaschev, M.: Which Kind of Module Should I Extract? In: Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK. vol. 477. CEUR-WS.org (2009)
27. Spackman, K.: Normal forms for description logic expressions of clinical concepts in SNOMED RT. In: AMIA 2001, American Medical Informatics Association Annual Symposium, Washington, DC, USA. AMIA (2001)
28. Spackman, K.A.: SNOMED RT and SNOMED CT. Promise of an international clinical ontology. M.D. computing : computers in medical practice **17**, 29 (2000)
29. The Gene Ontology Consortium: `http://geneontology.org/docs/go-subset-guide/`, Accessed: 2021-05-23
30. The Gene Ontology Consortium: The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Research **49**, gky1055 (2018)
31. Ulrich, B.: `https://bit.ly/2RqRA9A`, Accessed: 2021-06-06
32. Venkat-Raman, G., Tomson, C.R., Gao, Y., Cornet, R., Stengel, B., Gronhagen-Riska, C., Reid, C., Jacquelinet, C., Schaeffner, E., Boeschoten, E., Casino, F., Collart, F., De Meester, J., Zurriaga, O., Kramar, R., Jager, K.J., Simpson, K.: New primary renal diagnosis codes for the ERA-EDTA. Nephrology Dialysis Transplantation **27**(12), 4414–4419 (2012)

33. Vescovo, C.D., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Tsarkov, D.: Empirical Study of Logic-Based Modules: Cheap Is Cheerful. In: Proceedings of the Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia. vol. 8218, pp. 84–100. Springer (2013)
34. Wang, Z., Wang, K., Topor, R., Pan, J.: Forgetting for knowledge bases in DL-Lite. Annals of Mathematics and Artificial Intelligence **58**(1-2), 117–151 (2010)
35. Younesi, E., Malhotra, A., Gündel, M., Scordis, P., Kodamullil, A.T., Page, M., Müller, B., Springstubbe, S., Wüllner, U., Scheller, D., Hofmann-Apitius, M.: PDON: Parkinson's disease ontology for representation and modeling of the Parkinson's disease knowledge domain. Theoretical Biology & Medical Modelling (1), 1–17 (2015)
36. Zhao, Y., Schmidt, R.A.: On Concept Forgetting in Description Logics with Qualified Number Restrictions. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden. pp. 1984–1990. ijcai.org (2018)