Modularity Meets Forgetting: A Case Study with the SNOMED CT Ontology*

Jieying Chen,[1] Ghadah Alghamdi,[1] Renate A. Schmidt,[1]
Dirk Walther,[2] and Yongsheng Gao[3]

[1] The University of Manchester, UK
[2] DNV GL, Norway
[3] IHTSDO, UK

Abstract. Catering for ontology summary and reuse, several approaches such as modularisation and forgetting of symbols have been developed in order to provide users smaller sets of relevant axioms of an ontology. We consider different module extraction techniques and show how they relate to each other. We also consider the notion of uniform interpolation that is underlying forgetting. We show that significant improvements in the performance of forgetting can be obtained by applying a forgetting tool to ontology modules instead of the entire ontology. We investigate combining several module notions with uniform interpolation and provide a preliminary evaluation forgetting signatures based on the European Renal Association subset from SNOMED CT. Possible explanations for why modularity helps forgetting symbols from large-scale ontologies in practice are given. To facilitate the experiments, we develop a signature extension algorithm for the SNOMED CT ontology to additionally include more symbols relevant for users.

1 Introduction

SNOMED CT[1] is the most comprehensive, precise and widespread clinical ontology in the world with ample clinical specialties and requirements. The logic profile of SNOMED CT is a subset of the OWL 2 $\mathcal{EL}$ profile.[2] Description Logic with its reasoning capabilities plays an important role in the development and maintenance of SNOMED CT. The latest version of SNOMED CT from the year 2019 contains more than 340 000 axioms. The number of axioms increased by about 10% compared to the version from 2016.

SNOMED CT is still being developed and continuously updated. Maintaining and developing an ontology such as SNOMED CT is expensive and time-consuming. It is often more efficient for the ontology engineer to work with a subset of the ontology that contains all necessary information instead of the entire ontology. For example, the concept *kidney_disease (disorder)* has more than

* This work is partially funded by the EPSRC IAA 228 Project "Comparison and Abstraction of SNOMED CT Ontologies". We would like to thank Dr. Yizheng Zhao for helpful input on system FAME.

[1] `https://www.snomed.org`
[2] `https://www.w3.org/TR/owl2-profiles/#OWL_2_EL`

1 200 sub-concepts. When knowledge engineers redesign the concept model for a sub-hierarchy, it is useful to present developers a succinct sub-ontology to review and design the concept model.

A common use case for SNOMED CT is clinical data analytics. For instance, consider the scenario where the doctor would like to find diseases that have an inflammatory morphology and a finding site of kidney structure based on morphologies and/or finding sites. Instead of querying the whole of SNOMED CT, it would be more efficient to simply query a small subset of ontology containing the necessary axioms to preserve the relevant information.

Generally speaking, a module of an ontology is a subset of the ontology that can function in the same way as the original ontology in a specific context. This is formalised using a suitable inseparability relation. Model-theoretic and deductive inseparability relations have been studied. Several module notions based on inseparability relations have been proposed including plain, self-contained and depleting modules [10, 12]. The system MEX[3] has been implemented to extract minimal depleting and self-contained modules of acyclic $\mathcal{ELI}$-terminologies. Other notions are minimal subsumption modules [5, 7, 14], which are subsets of an ontology that preserve subsumption queries. The evaluation in [7] shows that minimal subsumption modules for $\mathcal{EL}/\mathcal{ELH}^r$-terminologies are generally much smaller than MEX-modules. However, deciding model-theoretic inseparability is expensive. The algorithm for computing minimal subsumption modules from $\mathcal{ELH}^r$-terminologies runs in exponential time. Approximate modules, such as locality-based modules [8] and other module extraction techniques via Datalog reasoning [21], can be computed rather efficiently. However, the resulting modules are not guaranteed to be minimal.

Concepts in the medical domain can be complicated to comprehend. Together with the fact that SNOMED CT contains more than 300 000 medical terms, it becomes clear that it can be very useful for ontology development or clinical data analytics to create an abstraction or summary of the ontology that only uses the terms that the developers are interested in.

Uniform interpolation and forgetting, as techniques of ontology abstraction, have attracted a lot of attention recently [25, 26]. Algorithms based on resolution have been developed for expressive description logics [15–17, 28, 30]. It has been shown that deciding the existence of uniform interpolants is 2-EXP-Complete for $\mathcal{ALC}$-TBoxes. Uniform interpolants do not always exist in $\mathcal{EL}$ and $\mathcal{ALC}$-TBoxes [10]. However, uniform interpolants always exist in DL-Lite ontologies [12]. Some approaches are proposed to compute uniform interpolants for lightweight description logics $\mathcal{EL}$ [11, 18]. Deciding existence of uniform interpolants in an $\mathcal{EL}$ ontology, such as SNOMED CT, is ExpTime-complete [19]. In the worst case, the size of uniform interpolants could be 3-EXP [20].

Given the high complexity result of finding uniform interpolants, in this paper, we are interested in computing uniform interpolants on SNOMED CT in practice. The signature in practice is usually much smaller than the number of symbols in the whole ontology, which means that the forgetting tool has to forget most of the symbols in the ontology. It is especially difficult to forget role names.

[3] `https://cgi.csc.liv.ac.uk/~konev/software/`

Precomputing ontology modules can help to reduce the number of symbols that need to be forgotten and also decrease the size of ontology, which motivates us to consider using modularity to approximate forgetting tools when computing uniform interpolants in practice.

This paper describes on-going work in a collaboration with IHTSDO about abstraction on the core version of SNOMED CT. In particular, we are interested in computing modules and uniform interpolants for smaller sets of concept names and role names. We first consider three different ontology modules and then analyse the relation among these modules. Then we give a brief overview of uniform interpolation/forgetting techniques and show the correctness of the optimization: speed up the forgetting process by precomputing ontology modules. We follow by proposing a signature extension method in SNOMED CT ontology. Our preliminary evaluation shows that precomputing subsumption modules significantly improves the performance of forgetting tools. Finally, we analyze the reasons why ontology modules can help optimize forgetting process.

2 Preliminaries

We start by recalling the definition of the description logic $\mathcal{EL}$ [2] and several of its extensions.

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be mutually disjoint and countably infinite sets of concept names and role names. The signature $\mathrm{sig}(\xi)$ is the set of concept and role names occurring in ξ, where ξ ranges over any syntactic object. The sets of *$\mathcal{EL}$-concepts* C, *$\mathcal{ELI}$-concepts* D, and the sets of *$\mathcal{ELH}$-axioms* α, *$\mathcal{ELI}$-axioms* β are built according to the grammar rules:

$$C ::= A \mid C \sqcap C \mid \exists r.C \qquad \alpha ::= C \sqsubseteq C \mid C \equiv C \mid r \sqsubseteq s$$
$$D ::= A \mid C \sqcap D \mid \exists r.D \mid \exists r^{-}.D \qquad \beta ::= D \sqsubseteq D \mid D \equiv D$$

where $A \in \mathsf{N_C}$ and $r, s \in \mathsf{N_R}$. An *$\mathcal{ELH}(\mathcal{ELI})$-TBox* is a finite set of $\mathcal{ELH}(\mathcal{ELI})$-axioms. A concept definition is an axiom of the form $C \equiv C$ or $D \equiv D$.

The semantics is defined as usual in terms of interpretations interpreting concept/role names and are then inductively extended to complex concepts. The notions of a model, satisfaction of a concept, axiom and TBox as well as the logical consequence relation are defined as usual; see, e.g., [3].

A terminology $\mathcal{T}$ is a TBox consisting of axioms such that the left-hand side of an axiom has to be a concept name, and no concept name occurs more than once on the left-hand side of an axiom.

An $\mathcal{EL}$-terminology $\mathcal{T}$ is *normalised* iff it only contains axioms of the forms $A \sqsubseteq B$, $A \sqsubseteq \exists r.C$, $\exists r.C \sqsubseteq A$ and $r \sqsubseteq s$, where $A, B \in \mathsf{N_C}$, $r \in \mathsf{N_R}$ and C is an $\mathcal{EL}$ concept.

For two general Tboxes $\mathcal{T}_1$ and $\mathcal{T}_2$, we say $\mathcal{T}_1$ and $\mathcal{T}_2$ are *Σ-inseparable*, denoted as $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$ if $\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{T}_1\} = \{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{T}_2\}$ [10].

3 Computing Ontology Modules

In this section, we consider three different module notions: locality-based modules [8], MEX-modules [10], and minimal subsumption modules [6, 7].

Locality-based Module. There exists three different types of syntactic locality-based modules, i.e., bottom ($\bot$), top ($\top$) and star ($\star$) modules. The latter combines the two former notions by iterative and exhaustive application.

MEX-Module. A MEX-module is a module extracted by the tool MEX for acyclic $\mathcal{ELI}$-terminologies. Intuitively, once removing the depleting module from an ontology, the remaining ontology states nothing about the signature and the symbols that are contained in the depleting module. A self-contained module is a sub-ontology that cannot be distinguished from the original ontology w.r.t. the signature and symbols in the module.

Definition 1 (Self-contained/Depleting Modules [10]). *Let $\mathcal{T}$ be an TBox and Σ a signature. Then $\mathcal{M} \subseteq \mathcal{T}$ is*

- *a* self-contained Σ-module *of $\mathcal{T}$ if $\mathcal{M} \equiv_{\Sigma \cup sig(\mathcal{M})} \mathcal{T}$;*
- *a* depleting Σ-module *of $\mathcal{T}$ if $\mathcal{T} \backslash \mathcal{M} \equiv_{\Sigma \cup sig(\mathcal{M})} \emptyset$.*

In case of acyclic $\mathcal{ELI}$-terminology $\mathcal{T}$, self-contained module and depleting module coincide, if $\mathcal{T}$ does not contain trivial concept definitions (cf. Theorem 29 [10]). So, a MEX-module is a minimal depleting module and self-contained module for acyclic $\mathcal{ELI}$-terminologies.

Minimal Subsumption Module. A subsumption module is a subset of an ontology that preserves subsumption queries that a user is interested in.

Definition 2 (Subsumption Module [7]). *Let $\mathcal{T}$ be an $\mathcal{ELH}$-terminology and let Σ be a signature. A subset $\mathcal{M} \subseteq \mathcal{T}$ is called an* $\mathcal{ELH}$-subsumption module of $\mathcal{T}$ w.r.t. Σ *iff for all $\mathcal{ELH}$-inclusions α with $sig(\alpha) \subseteq \Sigma$ it holds that $\mathcal{T} \models \alpha$ iff $\mathcal{M} \models \alpha$. $\mathcal{M}$ is called a* minimal subsumption module of $\mathcal{T}$ w.r.t. Σ *iff for any $\mathcal{M}' \subsetneq \mathcal{M}$, $\mathcal{M}'$ is not a subsumption module of $\mathcal{T}$ w.r.t. Σ.*

The following example with SNOMED CT shows the difference between the three different module notions. To simplify the presentation, we use A_1 to denote Neoplasm_uncertain_whether_benign_or_malignant, A_2 to denote Complex_mixed_AND/OR_stromal neoplasm, A to denote Mesoblastic_nephroma, B to denote Neoplasm, X to denote Neoplasm_and/or_hamartoma and Y to denote Tumor.

Example 1. Let $\Sigma = \{A, B\}$ and $\mathcal{T} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, where $\alpha_1 : A \sqsubseteq A_1 \sqcap A_2$, $\alpha_2 : A_1 \sqsubseteq B$, $\alpha_3 : A_2 \sqsubseteq B$ and $\alpha_4 : B \sqsubseteq X$. There are two subsumption modules of $\mathcal{T}$ w.r.t. Σ: $\{\alpha_1, \alpha_2\}$ and $\{\alpha_1, \alpha_3\}$. Either $\{\alpha_1, \alpha_2\}$ or $\{\alpha_1, \alpha_3\}$ is sufficient to preserve the entailment $A \sqsubseteq B$ that only uses symbols in Σ. The MEX-module and STAR-module of $\mathcal{T}$ w.r.t. Σ are each $\{\alpha_1, \alpha_2, \alpha_3\}$.

For $\mathcal{T}' = \{\alpha_1, \alpha_2, \alpha_3, \alpha_5\}$ with $\alpha_5 := B \equiv Y$,[4] the STAR-module w.r.t. Σ is $\mathcal{T}'$ itself. However, the minimal subsumption modules and MEX-module of $\mathcal{T}'$ w.r.t. Σ coincide with corresponding those of $\mathcal{T}$, respectively.

As mentioned in [10], the MEX-module of an $\mathcal{ELI}$-terminology w.r.t. Σ is always a subset of the respective STAR-module w.r.t. Σ. A STAR-module coincides with a MEX-module when the terminology contains no concept definitions (cf. Proposition 38 in [10]). Different to the notion of an MEX-module, which is defined in terms of the model-theoretic inseparability relation, the notion of a subsumption module is defined in terms of entailment queries. It is shown that a minimal subsumption module is contained in the respective MEX-module [4]. Hence, we can get the following proposition.

Proposition 1. *Let $\mathcal{T}$ be an $\mathcal{EL}$-terminology[5] and Σ a signature. Additionally, let $\mathcal{M}_\star$ and $\mathcal{M}_M$ be the STAR-module and the MEX-module of $\mathcal{T}$ w.r.t. Σ, respectively. Let $\mathcal{M}_S$ be a minimal subsumption module of $\mathcal{T}$ w.r.t. Σ. Then: $\mathcal{M}_S \subseteq \mathcal{M}_M \subseteq \mathcal{M}_\star$.*

Note that both module notions, MEX-modules and STAR-modules, each yield a unique subset of a given TBox w.r.t. a signature. On the other hand, there might exist several or even exponentially many minimal subsumption modules of $\mathcal{T}$ for a signature (cf. Example 6 in [7]).

There are two approaches for computing minimal subsumption modules: the glass-box and the black-box approach. In the glass-box approach, minimal subsumption modules are directly computed by combining subsumption justifications for every concept name in the signature [5]. The black-box approach uses a tool for detecting logical differences (e.g., CEX [9]) and computes the set of those axioms whose removal causes a logical difference w.r.t. the original ontology [6,7].

4 Computing Uniform Interpolants

The task of forgetting symbols from an ontology is also known as uniform interpolation. It can be used to reduce the amount of symbols in an ontology or hide certain confidential symbols without changing the meaning of the remaining symbols in the ontology. LETHE [13] and FAME [29] are two advanced tools for forgetting. While LETHE implements a resolution-based approach, FAME employs the Ackermann lemma [1] to perform the inferences during forgetting [27, 30]. We now give a formal definition of uniform interpolation.

Definition 3 (Uniform Interpolation). *Let $\mathcal{T}$ be an $\mathcal{EL}$-terminology and Σ a signature. U is a uniform interpolant of $\mathcal{T}$ w.r.t. Σ if the following conditions are satisfied:*

[4] Neoplasm (B) and Tumor (Y) are treated as synonyms in SNOMED CT and they share the same identifier. In order to illustrate the difference between MEX-module and STAR-module, we add α_5 in this example.

[5] Since MEX works for $\mathcal{ELI}$-terminologies and minimal subsumption modules are restricted to $\mathcal{ELH}$-terminologies, we consider $\mathcal{EL}$-terminologies here.

– *sig*$(U) \subseteq \Sigma$;
– *for every* $\mathcal{EL}$*-axiom* α *where* *sig*$(\alpha) \subseteq \Sigma$, $\mathcal{T} \models \alpha$ *iff* $U \models \alpha$.

As computing uniform interpolants is a difficult task especially for large-scale ontologies, the size and complexity of the input ontology influences the computation time directly. Instead of computing uniform interpolants on the whole ontology, we may be able to speed up the computation by computing uniform interpolants on ontology modules instead. The following proposition guarantees the correctness of this approach.

Proposition 2. *Let* $\mathcal{T}$ *be an* $\mathcal{EL}$*-TBox and* Σ *a signature. of* $\mathcal{T}$ *w.r.t.* Σ*. If* U *is a uniform interpolant of* $\mathcal{M}_S$*(*$\mathcal{M}_M$ *or* $\mathcal{M}_\star$*) w.r.t.* Σ*, then* U *is a uniform interpolant of* $\mathcal{T}$ *w.r.t.* Σ*.*

From Definition 2 and Definition 3, we can see that if U is a uniform interpolant of $\mathcal{M}_S$ w.r.t. Σ, then U is a uniform interpolant of $\mathcal{T}$ w.r.t. Σ. As $\mathcal{M}_S \subseteq \mathcal{M}_M$ by Proposition 1, we have that for every α where $\text{sig}(\alpha) \subseteq \Sigma$, if $\mathcal{M}_S \models \alpha$, then $\mathcal{M}_M \models \alpha$. Therefore, if U is a uniform interpolant of $\mathcal{M}_S$, then U is also a uniform interpolant of $\mathcal{M}_M$. Similarly, we have that if U is a uniform interpolant of $\mathcal{M}_S$, then U is a uniform interpolant of $\mathcal{M}_\star$.

5 Evaluation

We have evaluated the performance of the module extraction and forgetting tool on SNOMED CT. Considering that MEX only works on $\mathcal{ELI}$-terminologies and the module extraction tool of minimal subsumption module works on $\mathcal{ELH}^r$-terminologies, we choose to do the evaluation on an $\mathcal{EL}$-terminology fragment of SNOMED CT (version Jan 2016).[6] All the experiments were conducted on the machines equipped with Intel(R) Xeon(R) CPU E5-2640 v3 running at 2.60GHz and with 32GB RAM. The execution timeout was 1 hour. The forgetting tools we used in this experiment were FAME[7] and LETHE.[8]

5.1 Signature Extension

For the evaluation, we used the European Renal Association (ERA) subset of symbols from SNOMED CT which has been provided by IHTSDO (SNOMED International). The ERA-subset contains a list of primary renal diseases, which is designed specifically for use in renal centres and registries [23].

Previous evaluation of modularisation and forgetting tools on SNOMED CT typically involved computing random signatures, genuine seed signatures [24], or directly used SNOMED CT subsets [10]. However, these signatures do not properly reflect real-world scenarios of how users or developers would use ontology modules or uniform interpolants of the SNOMED CT ontology.

[6] $\mathcal{EL}$-terminology fragment is obtained by removing the axioms that are beyond $\mathcal{EL}$ profile. SNOMED CT (version Jan 2016) is a terminology itself.
[7] `http://www.cs.man.ac.uk/~schmidt/sf-fame/`
[8] `http://www.cs.man.ac.uk/~koopmanp/lethe/`

	$\#\mathsf{N_C}$	$\#\mathsf{N_R}$	$\#\mathcal{M}_\star$	$\#\mathcal{M}_M$	$\#\mathcal{M}_S$
$\Sigma^{\mathcal{T}}$	152	0	2078	861	152
Σ_0^+ / Σ_1^+	275	0	2078	861	391
Σ_2^+	369	10	2078	862	532

Table 1. Results of signature extension, size of different modules for extended signature and time for computing uniform interpolants

For example, the ERA-subset only contains a list of renal disorders. In order to relate symbols in the ERA-subset with symbols representing diseases, body structure, role names, etc., it is necessary to extend the signature. Intuitively, one could simply extend a signature based on the axioms in SNOMED CT. However, in practice the issue arises as to which axioms to choose and how much a signature should be extended. Based on discussion with developers from IHTSDO, we propose Algorithm 1 to extend a signature. In this algorithm, the function $\mathsf{role\_depth}()$ for $\mathcal{EL}$-concepts is recursively defined as follows:

$$\mathsf{role\_depth}(C) := \begin{cases} 0 & C \in \mathsf{N_C}; \\ \mathsf{max}(\mathsf{role\_depth}(D), \mathsf{role\_depth}(E)) & C = D \sqcap E; \\ 1 + \mathsf{role\_depth}(D) & C = \exists r.D. \end{cases}$$

In version Jan 2016 of SNOMED CT, the deepest role depth of any complex concept is 2.

Algorithm 1 extends the original signature Σ in three different ways. Briefly speaking, for every concept name $A \in \Sigma^{\mathcal{T}}$, Σ_i^+ includes all A's direct superconcepts C where role depth of C is less than i, where $0 \leq i \leq 2$.

As we can see in Table 1, the size of signature increases from 152 ($|\Sigma^{\mathcal{T}}|$) to 275 ($|\Sigma_0^{\mathcal{T}}|$). However, neither a concept name nor a role name was added when extended signature from $\Sigma_0^{\mathcal{T}}$ to $\Sigma_1^{\mathcal{T}}$. Another 10 role names and 94 con-

Algorithm 1 Signature-Extension($\mathcal{T}$, Σ)

Input: Normalised Terminology $\mathcal{T}$, Signature Σ
Output: Σ_0^+, Σ_1^+, Σ_2^+

1: $\Sigma^{\mathcal{T}} := \Sigma \cap \mathrm{sig}(\mathcal{T}) \cap \mathsf{N_C}$
2: $\Sigma_0^+ := \Sigma_1^+ := \Sigma_2^+ := \Sigma^{\mathcal{T}}$
3: **for** $A \sqsubseteq C \in \mathcal{T}$ with $A \in \Sigma^{\mathcal{T}}$ **do**
4: **if** $C \in \mathsf{N_C}$ **then**
5: $\Sigma_0^+ := \Sigma_0^+ \cup \{C\}$
6: **if** $\mathsf{role\_depth}(C) = 1$ **then**
7: $\Sigma_1^+ := \Sigma_1^+ \cup \mathrm{sig}(C)$
8: **if** $\mathsf{role\_depth}(C) = 2$ **then**
9: $\Sigma_2^+ := \Sigma_2^+ \cup \mathrm{sig}(C)$

Tool	FAME			LETHE		
$\mathcal{T}$	$\mathcal{M}_\star$	$\mathcal{M}_M$	$\mathcal{M}_S$	$\mathcal{M}_\star$	$\mathcal{M}_M$	$\mathcal{M}_S$
Min.	0.25	0.22	0.22	0.55	0.39	0.47
Max.	7.70	871.00	2.00	0.55	0.81	2.60
Avg.	2.20	387.00	0.41	0.55	0.56	0.83
Med.	0.36	306.00	0.26	0.55	0.56	0.62
Succ.	4/14	11/14	13/14	1/14	12/14	14/14

Table 2. Computation time (s) of uniform interpolants on different modules by FAME and LETHE using 14 signatures

cept names were further included in $\Sigma_2^{\mathcal{T}}$. About the size of modules, the size of STAR-modules stays the same. This is due to the fact that the signature is extended when computing STAR-modules. The size of MEX-module stays almost the same. But the signature extension causes an impact on the size of minimal subsumption modules.

5.2 Signature Partition

The second experiment is designed to mimic the scenario where a user queries the ontology. In this case, only a rather small number of closely related concept and role names are expected to be contained in the signature. To obtain such small signatures, we devised Algorithm 2 performing signature partitioning. Function $\mathsf{ExtractStarModule}(\Sigma, \mathcal{T})$ in Line 1 is provided by the OWL API[9] for computing STAR-modules. Even though the function accepts general TBoxes $\mathcal{T}$ formulated in OWL2, we only use it as a convenient way to compute the module $\mathcal{M}_\star$ of SNOMED CT for the ERA concept names as a signature. The function $\mathsf{Classify}(\mathcal{M}_\star)$ in Line 2 then calls the reasoner ELK[10] to classify $\mathcal{M}_\star$. In a subsequent step, we reduce the computed class hierarchy to the symbols in the input signature. We obtain a classification $\mathcal{H}'$ of the ERA concept names. The method $\mathsf{Partition}(\mathcal{M}^\star)$ in Line 3 involved user interaction at the time of its conceptualisation. Later it became clear that this step could also be automated. We first display $\mathcal{H}'$ using an ontology visualisation tool, e.g., WebVOWL.[11] Then we partition $\mathcal{H}'$ by identifying different sets $\mathcal{H}'_i$ of concept inclusions such that every pair of concept names from the same set $\mathcal{H}'_i$ are connected via a chain of concept inclusions from $\mathcal{H}'_i$, and every pair of concept names taken from different sets $\mathcal{H}'_i$ and $\mathcal{H}'_j$ with $i \neq j$ are not connected in this sense. This results in 14 disjoint sub-hierarchies: $\mathcal{H}'_1, \mathcal{H}'_2, ..., \mathcal{H}'_{14}$. The loop from Line 4 to 6, first computes the signature Σ_i of hierarchy $\mathcal{H}'_i$ and then extends Σ_i to Σ_i^+ using function $\mathsf{Signature\text{-}Extension}(\mathcal{M}_\star, \Sigma_i)$ presented in Algorithm 2. The resulting 14 signatures consist of 5 to 40 concept names and 0 to 8 role names.

[9] `http://owlapi.sourceforge.net/`
[10] `https://www.cs.ox.ac.uk/isg/tools/ELK/`
[11] `http://vowl.visualdataweb.org/webvowl.html`

Algorithm 2 Signature-Partitioning($\mathcal{T}$, Σ)

Input: Terminology $\mathcal{T}$, Signature Σ (ERA subset)
Output: extended signatures $\Sigma_1^+, ..., \Sigma_n^+$

1: $\mathcal{M}_\star := \mathsf{ExtractStarModule}(\Sigma, \mathcal{T})$
2: $\mathcal{H}' := \mathsf{Reduce}(\Sigma, \mathsf{Classify}(\mathcal{M}_\star))$
3: $\langle \mathcal{H}'_1, ..., \mathcal{H}'_n \rangle := \mathsf{Partition}(\mathcal{H}')$
4: **for** $i \in \{1, ..., n\}$ **do**
5: $\quad \Sigma_i := \mathrm{sig}(\mathcal{H}'_i)$
6: $\quad \Sigma_i^+ := \mathsf{Signature\text{-}Extension}(\mathcal{M}_\star, \Sigma_i)$

For each of the 14 signatures, we computed three different types of modules of SNOMED CT: STAR-module, MEX-module and the minimal subsumption modules. These modules together with their respective producing signatures were then taken as input for the systems FAME and LETHE to compute uniform interpolants. The resulting computation times are summarised in Table 2 in terms of the mininal, maximal, average, and median time taken to compute a uniform interpolant. Only the successful cases that finished within a timeout of 1 hour were counted. For example, the min/max/avg/med values in Column 5 are the same since LETHE managed to compute a uniform interpolant for only one out of 14 signatures.

It becomes evident in Table 2 that precomputing MEX-modules and minimal subsumption modules significantly reduces the computation time and, thus, increases the success rate of computing uniform interpolants for both tools to more than 90%. In particular in the case of minimal subsumption modules, the uniform interpolant for any signature can be computed within 2.6 seconds by LETHE. Contrast this with the fact that LETHE takes for all but one signature more than one hour when using STAR-modules as an input.

However, we need to keep in mind that computing minimal subsumption modules can be computationally expensive as well. The time needed to compute the minimal subsumption modules for the 14 signatures ranged from 1 to 939 seconds. The alternative seems to be the use of MEX-modules as they can be computed even more efficiently. On the other hand, not all MEX-modules reduced the computation time for uniform interpolants to less than one hour, cf. Table 2. Hence, depending on the timeout constraint, the use of minimal subsumption modules can enable the computation of uniform interpolants.

Table 6 shows the size of uniform interpolants for three different types of modules w.r.t. the signatures for which all uniform interpolants can be successfully computed. In general, the uniform interpolants for the different modules are rather similar in size.

	$\lvert\#\text{sig}_{\mathsf{N_C}}(\mathcal{M}_\star)$	$\#\text{sig}_{\mathsf{N_C}}(\mathcal{M}_M)$	$\#\text{sig}_{\mathsf{N_C}}(\mathcal{M}_S)$
Min.	7	6	6
Max.	319	172	95
Avg.	97.9	47.1	24.4
Med.	96	41	14.5

Table 3. Number of concept names in different modules

	$\lvert\#\text{sig}_{\mathsf{N_R}}(\mathcal{M}_\star)$	$\#\text{sig}_{\mathsf{N_R}}(\mathcal{M}_M)$	$\#\text{sig}_{\mathsf{N_R}}(\mathcal{M}_S)$
Min.	0	0	0
Max.	17	15	8
Avg.	5	4	3.5
Med.	4	3.5	3.5

Table 4. Number of role names in different modules

	$\#\mathcal{M}_\star$	$\#\mathcal{M}_M$	$\#\mathcal{M}_S$
Min.	6	2	2
Max.	304	158	64
Avg.	92.7	41.8	14.8
Med.	91	38	6.5

Table 5. Number of axioms in different modules

	$\lvert\#U(\mathcal{M}_\star)$	$\#U(\mathcal{M}_M)$	$\#U(\mathcal{M}_S)$
Min.	6	5	5
Max.	14	25	37
Avg.	10.5	11.7	12.7
Med.	11	11	9

Table 6. Number of axioms in uniform interpolant for different modules

6 Discussion

The results in Section 5 show that precomputing minimal subsumption modules and MEX-modules can significantly speed up the process of computing uniform interpolants. In this section, we analyse the reasons why module extraction techniques can help to approximate forgetting tools.

Smaller module. Table 5 shows that, on average, the size of minimal subsumption modules is almost 2 times smaller than MEX-modules, and 5 times smaller than STAR-modules (even 13 times smaller than STAR-module according to median value). Example 1 illustrates why minimal subsumption modules are smaller than MEX-modules and STAR-modules.

Fewer symbols to forget. As we can see from Table 3, the number of concept names that occur in minimal subsumption modules is 53% less than MEX-modules and almost 3 times less than STAR-modules. As the interpolation signature is the same for all modules, forgetting on subsumption modules has much fewer concept names to forget, the same as MEX-modules. Although forgetting role names is more difficult than forgetting concept names, the number of role names does not vary much on average for the different modules, cf. Table 4.

Special role "RoleGroup". In SNOMED CT, a special role name, called "RoleGroup", maintains correct inferences and semantic meaning for complex concept expressions that relate to, e.g., multiple sites and morphologies [22].

In our preliminary research, it is found that the presence of "RoleGroup" makes the forgetting problem harder. However, inspection has revealed that "RoleGroup" is occurred less frequently in minimal subsumption modules.

Influence of new types of axiom. Although precomputing MEX-modules and minimal subsumption modules can remarkably speed up the forgetting process, these techniques apply to terminologies with few operators.

$\#(r \sqsubseteq s)$	$\#(C \sqsubseteq A)$	$\#(r \circ s \sqsubseteq r)$	$\#r^-$	$\#r^+$	$\#\text{Diff}(\mathcal{H}, \mathcal{H}_{\mathcal{ELH}})$	$\#\text{Diff}(\mathcal{H}, \mathcal{H}_{\mathcal{EL}})$
123	20	5	4	2	2558	25973

Table 7. Classification result on $\mathcal{EL}$,$\mathcal{ELH}$ and whole SNOMED CT

The latest version of SNOMED CT (Version Jan 2019) included new types of axioms compared with Version Jan 2016, which is beyond $\mathcal{ELH}$-terminologies. That is, property chain of the form $r \circ s \sqsubseteq r$, reflexive property(r^-), transitive property(r^+), concept inclusion where the left-hand side is a complex concept. Another type of axiom we also have to consider is property inclusion that MEX cannot deal with.

With the purpose of figuring out how these axioms influence the subsumption result in SNOMED CT, we conducted the following experiment. First, we got the $\mathcal{ELH}$-Terminology fragment of SNOMED CT (Version Jan 2019), denoted as $\mathcal{T}_{\mathcal{ELH}}$, by removing axioms of property chain, reflexive property, transitive property and subClass axioms in the form of $C \sqsubseteq A$. The $\mathcal{EL}$-fragment of SNOMED CT, denoted as $\mathcal{T}_{\mathcal{EL}}$, is extracted by further removing property inclusion from $\mathcal{T}_{\mathcal{ELH}}$. We then employed the ELK reasoner to classify on the original SNOMED CT, $\mathcal{T}_{\mathcal{EL}}$, $\mathcal{H}_{\mathcal{ELH}}$ and get classification ontologies $\mathcal{O}$, $\mathcal{O}_{\mathcal{ELH}}$ and $\mathcal{O}_{\mathcal{EL}}$. The function $\text{Diff}(\mathcal{H}, \mathcal{H}')$ compares the set difference between two classified ontologies. As shown in Table 7, the entailment difference between $\mathcal{H}$ and $\mathcal{H}_{\mathcal{EL}}$ is 25973. There is around 2500 entailments difference between $\mathcal{H}$ and $\mathcal{H}_{\mathcal{ELH}}$. This means role inclusions do have a considerable effect on SNOMED CT. Although MEX is very efficient at extracting modules, MEX-modules are likely to lose some relevant information about the signature when extracting modules only on $\mathcal{EL}$-fragment of SNOMED CT, similar as extracting minimal subsumption modules just on $\mathcal{ELH}$-fragment of SNOMED CT.

7 Conclusion and Future Work

In this paper, we first briefly described current techniques on ontology modularization and uniform interpolation. We provided the preliminary evaluation of current techniques on comprehensive medical ontology SNOMED CT for a set of meaningful signatures in practice. In future, we plan to further investigate the performance of these techniques on real-world signatures. We also expect to evaluate the quality of modules and uniform interpolants we obtained with the developers of SNOMED CT and, ideally, also with doctors and nurses. For better use of current module extraction and uniform interpolation techniques in real-world situation, we will update current module/UI techniques according to feedback from SNOMED CT developers. Besides, the algorithm for computing minimal subsumption modules is also expected to be updated in order to deal with logic profiles that are outside its current scope.

References

1. Ackermann, W.: Untersuchungen $\ddot{u}$ber das Eliminationsproblem der mathematischen Logik. Mathematische Annalen 110(1), 390–413 (1935)
2. Baader, F.: Terminological cycles in a description logic with existential restrictions. In: Proc. of IJCAI'03. pp. 325–330 (2003)
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
4. Chen, J.: Knowledge Extraction from Description Logic Terminologies. Ph.D. thesis, University of Paris-Saclay, France (2018)
5. Chen, J., Ludwig, M., Ma, Y., Walther, D.: Zooming in on ontologies: Minimal modules and best excerpts. In: Proc. of ISWC'17. pp. 173–189 (2017)
6. Chen, J., Ludwig, M., Walther, D.: On computing minimal EL-subsumption modules. In: Proc. of WOMoCoE'16 (2016)
7. Chen, J., Ludwig, M., Walther, D.: Computing minimal subsumption modules of ontologies. In: Proc. of GCAI'18. pp. 41–53 (2018)
8. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research 31(1), 273–318 (2008)
9. 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)
10. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artificial Intelligence 203, 66–103 (2013)
11. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: In Prof. of IJCAI 2009. pp. 830–835 (2009)
12. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to dl-lite. Artif. Intell. 174(15), 1093–1141 (2010)
13. Koopmann, P., Schmidt, R.A.: LETHE: A saturation-based tool for non-classical reasoning. In: Proc. of ORE'15 (2015)
14. Koopmann, P., Chen, J.: Computing ALCH-subsumption modules using uniform interpolation. In: Proc. of SOQE'17. pp. 51–66 (2017)
15. Koopmann, P., Schmidt, R.A.: Forgetting Concept and Role Symbols in $\mathcal{ALCH}$-Ontologies. In: Proc. of LPAR'13. LNCS, vol. 8312, pp. 552–567 (2013)
16. Koopmann, P., Schmidt, R.A.: Uniform Interpolation of ALC-Ontologies Using Fixpoints. In: Proc. FroCoS'13. LNCS, vol. 8152, pp. 87–102 (2013)
17. Koopmann, P., Schmidt, R.A.: Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies. In: Proc. IJCAR'14. LNCS, vol. 8562, pp. 434–448 (2014)
18. Ludwig, M., Walther, D.: Towards a practical decision procedure for uniform interpolants of el-tboxes - a proof-theoretic approach. In: Proc. of GCAI'16. pp. 147–160 (2016)
19. Lutz, C., Seylan, I., Wolter, F.: An automata-theoretic approach to uniform interpolation and approximation in the description logic EL. In: Proc. KR'12 (2012)
20. Nikitina, N., Rudolph, S.: Expexpexplosion: Uniform interpolation in general EL terminologies. In: Proc. of ECAI 2012. pp. 618–623 (2012)
21. Romero, A.A., Kaminski, M., Grau, B.C., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. Journal of Artificial Intelligence Research 55, 499–564 (2016)
22. Spackman, K.A., Dionne, R., Mays, E., Weis, J.: Role grouping as an extension to the description logic of ontylog, motivated by concept modeling in SNOMED. In: Proc. of AMIA'02 (2002)

23. Venkat-Raman, G., Boeschoten, E., Casino, F., Collart, F., De Meester, J., Zurriaga, O., Kramar, R., Simpson, K., Tomson, C.R., Gao, Y., Cornet, R., Jager, K.J., Stengel, B., Gronhagen-Riska, C., Reid, C., Jacquelinet, C., Schaeffner, E.: New primary renal diagnosis codes for the ERA-EDTA. Nephrology Dialysis Transplantation 27(12), 4414–4419 (2012)
24. Vescovo, C.D., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Tsarkov, D.: Empirical study of logic-based modules: Cheap is cheerful. In: Proc. of DL'13. pp. 144–155 (2013)
25. Wang, K., Wang, Z., Topor, R., Pan, J.Z., Antoniou, G.: Eliminating concepts and roles from ontologies in expressive description logics. Computational Intelligence 30(2), 205–232 (2014)
26. Zhang, X., Lin, Z., Wang, K.: A tableau algorithm for paraconsistent and nonmonotonic reasoning in description logic-based system. In: Proc. of APWeb'11. pp. 345–356 (2011)
27. Zhao, Y., Alghamdi, G., Schmidt, R.A., Feng, H., Stoilos, G., Juric, D., Khodadadi, M.: Tracking logical difference in large-scale ontologies: A forgetting-based approach. In: Proc. of AAAI'19 (2019)
28. Zhao, Y., Schmidt, R.A.: Forgetting Concept and Role Symbols in $\mathcal{ALCOIH}\mu^+(\nabla, \sqcap)$-Ontologies. In: Proc. of IJCAI'16. pp. 1345–1352 (2016)
29. Zhao, Y., Schmidt, R.A.: FAME: an automated tool for semantic forgetting in expressive description logics. In: Proc. of IJCAR'18. pp. 19–27 (2018)
30. Zhao, Y., Schmidt, R.A.: On Concept Forgetting in Description Logics with Qualified Number Restrictions. In: Proc. of IJCAI'18. pp. 1984–1990 (2018)