

# Ontology Extraction for Large Ontologies via Modularity and Forgetting

Jieying Chen  
Univ. of Manchester, UK  
Univ. of Oslo, Norway

Ghadah Alghamdi  
Renate A. Schmidt  
Univ. of Manchester, UK

Dirk Walther  
DNV GL  
Oslo, Norway

Yongsheng Gao  
IHTSDO  
London, UK

## ABSTRACT

We are interested in the computation of ontology extracts based on forgetting from large ontologies in real-world scenarios. Such scenarios require nearly all of the terms in the ontology to be forgotten, which poses a significant challenge to forgetting tools. In this paper we show that modularization and forgetting can be combined beneficially in order to compute ontology extracts. While a module is a subset of axioms of a given ontology, the solution of forgetting (also known as a uniform interpolant) is a compact representation of the ontology limited to a subset of the signature. The approach introduced in this paper uses an iterative workflow of four stages: (i) extension of the given signature and, if needed partitioning, (ii) modularization, (iii) forgetting, and (iv) evaluation by domain expert. For modularization we use three kinds of modules: locality-based, semantic and minimal subsumption modules. For forgetting three tools are used: NUI, LETHE and FAME. An evaluation on the SNOMED CT and NCI ontologies for standard concept name lists showed that precomputing ontology modules reduces the number of terms that need to be forgotten. An advantage of the presented approach is high precision of the computed ontology extracts.

## CCS CONCEPTS

• **Computing methodologies** → **Description logics**; *Knowledge representation and reasoning*; *Ontology engineering*; • **Theory of computation** → *Abstraction*.

## KEYWORDS

Knowledge management, knowledge representation and reasoning, knowledge abstraction, ontology abstraction, description logics, ontology modularity, uniform interpolation, forgetting

### ACM Reference Format:

Jieying Chen, Ghadah Alghamdi, Renate A. Schmidt, Dirk Walther, and Yongsheng Gao. 2019. Ontology Extraction for Large Ontologies via Modularity and Forgetting. In *10th International Conference on Knowledge Capture (K-CAP '19)*, November 19–21, 2019, Marina Del Rey, CA, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3360901.3364424>

## 1 INTRODUCTION

The creation of ontology extracts is an essential operation for the reuse, creation, evaluation, curation, decomposition, integration

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

K-CAP '19, November 19–21, 2019, Marina Del Rey, CA, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7008-0/19/11.

<https://doi.org/10.1145/3360901.3364424>

and general use of ontologies. For example, for reviewing and analyzing the information relating to the concept *kidney\_disease (disorder)*, which has more than 1 200 sub-concepts in a medical ontology such as the SNOMED CT ontology, developers would benefit from being able to work with an extract that succinctly summarizes all information in the ontology relating to kidney diseases. SNOMED CT contains more than 340 000 axioms and about as many concepts (Jan. 2019 version). Another concrete scenario is a doctor wishing to find diseases with an inflammatory morphology and a finding site of kidney structure based on morphologies and/or finding sites. Instead of querying the ontology as a whole, it would be more efficient to simply query a smaller extract of the ontology containing sufficiently many axioms to compute the same answer as if it was computed for the entire ontology.

Ontology modularity has been developed to tackle these challenges of reuseability and interoperability [5, 12, 16, 18, 30, 39]. Modularization creates a slice of an ontology for an input (seed) signature. The idea is that all axioms in the ontology are returned which contain information relevant to the input signature. Among the different modularization approaches locality-based modularization is frequently used to extract subsets of an ontology for further localized, and often easier, processing with other tools, such as reasoning, querying, retrieval and ontology mapping [8, 13, 14]. However, empirical investigations [16, 25, 41] in application-close scenarios involving SNOMED CT have found that graph-based approaches to modularization have reasonable coverage (71%–96%), but the obtained extracts are large (17%–51% of the size of the ontology). While relatively small extracts can be obtained with locality-based modules, a down-side is lower precision due to the presence of a large number of symbols in the module outside the desired signature specified as input.

An alternative method for ontology extraction is *forgetting*, also known as *uniform interpolation* [9, 17, 18, 23, 24, 43], not to be confused with related notions of interpolation as in, e.g., [35]. Forgetting creates a compact representation of a part of the information contained in an ontology that preserves the underlying logical definitions of the specified terms (the *interpolation signature*)<sup>1</sup> by hiding the remaining terms. This allows users to focus exactly on the information they are interested in. (By ‘terms’ we mean concept and role names.) Applications of forgetting include *ontology reuse* taking advantage of the potentially much smaller result of forgetting, *predicate hiding* concealing confidential terms, *logical difference* facilitating *ontology versioning*, and *ontology summary* providing a focused extract facilitating ontology comprehension.

In this paper we refer to ontology extracts computed by forgetting as *uniform interpolants (UIs)*. UIs have high precision, but many of their axioms do not belong to the original ontology because they

<sup>1</sup>We refer signature as interpolation signature in the rest of the paper.

are inferred axioms. In contrast, modules contain only axioms from the original ontology but typically contain many terms outside the specified interpolation signature.

Because it uses reasoning, forgetting is generally harder than standard reasoning and modularization. Despite the high computational complexity there has been recent progress in developing forgetting methods with feasible performance on small to medium size ontologies for randomly generated signatures [17, 21, 26, 45, 46]. Current forgetting tools struggle however on large ontologies such as SNOMED CT for *real-world signatures*, which tend to be very small compared with the signature of the whole ontology. The smaller the interpolation signature, the more work forgetting tools have in order to compute UIs.

The main contribution of the paper is a *novel workflow consisting of four stages: (a) signature extension, (b) ontology module extraction, (c) forgetting, and (d) feedback by domain experts, which is evaluated on the SNOMED CT and NCIt ontologies*. Our investigation uses three different modularization approaches (locality-based, semantic and minimal subsumption modularization) and three forgetting tools (NUI, LETHE and FAME).

A central component of the workflow is a *signature extension algorithm developed with input from domain experts*, because the shape and quality of ontology modules and UIs depend heavily on the input signature. It is unrealistic to expect that users have theoretical background and modularization/forgetting tool developers have domain knowledge to be able to provide suitable input signatures. For our target ontologies there exist standard lists of concept names, e.g., for SNOMED CT these are called *refsets*. These sets support clinicians by facilitating the search for information on particular, closely related terms. We present an algorithm that utilizes the ontology and existing refsets for computing suitable input interpolation signatures. The modules extracted in the second stage preserve all subsumption queries over the input signature. We prove that computing a UI from such modules preserves these queries as well, and is therefore logically equivalent to a UI computed from the original ontology without using modularization. This shows the correctness of the workflow.

The extracts computed by our workflow have been found to be useful by domain experts. A crucial enrichment of the user-defined signature is provided resulting in more comprehensible UIs. Samples of our UIs for various sizes and all experimental data can be accessed under <http://bit.ly/2JEearaz>.

The paper is structured as follows. In Section 2 we describe the ontologies targeted by our research and recall definitions of basic notions of description logics. Sections 3 and 4 give background on ontology modularity and forgetting. Section 5 presents the workflow and shows correctness of combining forgetting with modularization. Evaluation results of our method on the SNOMED CT and NCIt ontologies for real-world scenarios are presented in Section 6. In Section 7 we discuss reasons for why ontology modules help to improve the forgetting process and finish with a conclusion.

## 2 TARGET ONTOLOGIES

The target ontologies of the present research are SNOMED CT and NCIt.

SNOMED CT<sup>2</sup> is a comprehensive and widely-used medical ontology covering various clinical specialities and requirements [36, 37]. It provides definitions of standard medical terminologies used in health records for the purpose of supporting interoperability between systems used in health care services in several countries. SNOMED CT is actively managed, curated and distributed by the International Health Terminology Standards Development Organisation (IHTSDO). The logic profile of SNOMED CT is effectively an  $\mathcal{ELH}$ -TBox in a subset of the OWL 2  $\mathcal{EL}$  profile. Axioms in SNOMED CT are represented in OWL 2 expressions.<sup>3</sup> The new property axioms, such as reflexive roles, transitive roles, and role chains were introduced in July 2018. The acyclic  $\mathcal{EL}$ -fragment of SNOMED CT contains more than 99.9% of the original ontology. It contains more than 349 000 axioms, with the number of axioms increasing by about 10% compared to the version from January 2016. A challenge of extracting knowledge from SNOMED CT is that the axioms in SNOMED CT typically contain nested existential restrictions, which makes reasoning more intricate.

The NCIt<sup>4</sup> ontology is a thesaurus of biomedical terminology covering different cancer-related information [11]. The thesaurus is updated monthly and provides a range of vocabularies including cancer terms, a drug dictionary and genetic terms. The latest release 19.06d includes more than 200 000 axioms. The DL expressivity of the NCIt is  $\mathcal{SH}$  ( $\mathcal{ALC}$  with role hierarchies and transitive roles), though more than 99.9 % axioms are formulated in  $\mathcal{EL}$ . There is no nesting of existential restrictions in the axioms.

For the evaluation we have reduced SNOMED CT and NCIt to axioms in the description logic  $\mathcal{EL}$ . Let  $N_C$  and  $N_R$  be mutually disjoint and countably infinite sets of concept names and role names. The signature  $\text{sig}(\xi)$  is the set of concept and role names occurring in  $\xi$ , where  $\xi$  ranges over any syntactic object or ontology. The sets of  $\mathcal{EL}$ -concepts  $C$  and  $\mathcal{EL}$ -axioms  $\alpha$  are built according to the grammar rules:  $C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$  and  $\alpha ::= C \sqsubseteq C \mid C \equiv C$ , where  $A \in N_C$  and  $r \in N_R$ , while the sets of  $\mathcal{ALC}$ -concepts  $D$  and  $\mathcal{ALC}$ -axioms  $\beta$  are built in accordance with the rules:

$$D ::= \top \mid \perp \mid A \mid D \sqcap D \mid \exists r.D \mid D \sqcup D \mid \neg D \mid \forall r.D$$

$$\beta ::= D \sqsubseteq D \mid D \equiv D.$$

An  $\mathcal{EL}(\mathcal{ALC})$ -TBox is a finite set of  $\mathcal{EL}(\mathcal{ALC})$ -axioms. An  $\mathcal{ELH}$ -TBox is an  $\mathcal{EL}$ -TBox additionally allows role inclusion ( $r \sqsubseteq s$ ). An  $\mathcal{ELH}^r$ -TBox further allows range restriction.

The semantics is defined as usual [1]. Concepts and roles are interpreted as sets and binary relations respectively, the Boolean operators as the corresponding set operations, the existential restriction operator  $\exists$  as the pre-image operation, the inclusion relation  $\sqsubseteq$  as the subset relationship and mutual subsumption  $\equiv$  as equivalence. The notions of satisfaction of a concept, axiom and TBox as well as a model and logical consequence are defined as expected.

An  $\mathcal{L}$ -terminology  $\mathcal{T}$  is an  $\mathcal{L}$ -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. In this paper, we consider  $\mathcal{L}$  to range over  $\mathcal{EL}$ ,  $\mathcal{ELH}$ ,  $\mathcal{ELH}^r$ , and  $\mathcal{ALC}$ . A terminology is *acyclic* if no concept name is defined in terms of itself. SNOMED CT is acyclic, whereas NCIt

<sup>2</sup><https://www.snomed.org>

<sup>3</sup><http://snomed.org/owl>

<sup>4</sup><https://ncit.nci.nih.gov>

contains cyclic dependencies. An  $\mathcal{EL}$ -terminology  $\mathcal{T}$  is *normalised* iff it only contains axioms of the forms  $A \sqsubseteq C$ ,  $A \sqsubseteq \exists r.C$ , and  $\exists r.C \sqsubseteq A$ , where  $A \in \mathcal{N}_C$ ,  $r \in \mathcal{N}_R$  and  $C$  is an  $\mathcal{EL}$ -concept.

### 3 ONTOLOGY MODULARITY

In the area of description logics, a *module* of an ontology is a subset of the ontology for which answers for queries over the input signature  $\Sigma$  are the same as those obtained from the ontology. We consider three types of modules: semantic modules, minimal subsumption modules and locality-based modules.

*Semantic modules* are logic-based modules whose formalization uses the model-theoretic inseparability relation  $\equiv_\Sigma$  [15, 16]. Two general TBoxes  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are  $\Sigma$ -inseparable, written  $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$ , if  $\{\mathcal{I}_\Sigma \mid \mathcal{I} \models \mathcal{T}_1\} = \{\mathcal{I} \mid \mathcal{I} \models \mathcal{T}_2\}$ . Three different semantic modules exist:

*Definition 3.1 (Semantic modules [15, 16]).* Let  $\mathcal{T}$  be an  $\mathcal{L}$ -TBox and let  $\Sigma$  be a signature. Then  $\mathcal{M} \subseteq \mathcal{T}$  is

- a *plain  $\Sigma$ -module* of  $\mathcal{T}$  if  $\mathcal{M} \equiv_\Sigma \mathcal{T}$ ,
- a *self-contained  $\Sigma$ -module* of  $\mathcal{T}$  if  $\mathcal{M} \equiv_{\Sigma \cup \text{sig}(\mathcal{M})} \mathcal{T}$ , and
- a *depleting  $\Sigma$ -module* of  $\mathcal{T}$  if  $\mathcal{T} \setminus \mathcal{M} \equiv_{\Sigma \cup \text{sig}(\mathcal{M})} \emptyset$ .

When a depleting module for a signature is being removed from an ontology, the remaining ontology states nothing about the signature and the additional symbols that are contained in the depleting module. In the case of  $\mathcal{ELI}$ -TBoxes (with inverse roles), the notions of self-contained  $\Sigma$ -module and depleting  $\Sigma$ -module coincide, if  $\mathcal{T}$  does not contain trivial concept definitions (cf. Theorem 29 [16]). The *MEX system*<sup>5</sup> extracts minimal depleting and self-contained semantic modules from ontologies formulated as  $\mathcal{ELI}$ -terminologies [16].

The recently introduced (*minimal*) *subsumption modules* preserve subsumption queries [4, 7, 19].

*Definition 3.2 (Subsumption modules [7]).* Let  $\mathcal{T}$  be an  $\mathcal{L}$ -TBox and let  $\Sigma$  be a signature. A subset  $\mathcal{M}$  of  $\mathcal{T}$  is called an  *$\mathcal{L}$ -subsumption module* of  $\mathcal{T}$  w.r.t.  $\Sigma$  iff for all  $\mathcal{L}$ -inclusions  $\alpha$  with  $\text{sig}(\alpha) \subseteq \Sigma$  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.  $\Sigma$  iff for any  $\mathcal{M}' \subsetneq \mathcal{M}$ ,  $\mathcal{M}'$  is not a subsumption module of  $\mathcal{T}$  w.r.t.  $\Sigma$ .

Evaluation has shown that minimal subsumption modules are generally much smaller than semantic modules [7]. However, deciding the preservation of subsumption queries can be expensive. For  $\mathcal{ELH}^r$ -terminologies the algorithm for computing minimal subsumption modules runs in exponential time.

Approximate modules, such as modules based on syntactic locality [12], modules extracted via Datalog reasoning [33] and reachability modules [32] can be computed efficiently. In the present research we have used syntactic *locality-based modularization* for which polynomial time algorithms exist. There are three different types of syntactic locality-based modules, i.e., bottom ( $\perp$ ), top ( $\top$ ) and star ( $\star$ ) modules [12]. The latter combines the two former notions by iterative and exhaustive application. In our evaluation we used star modularization available as part of the OWL API. Star modules tend to be smaller than their bottom and top counterparts and still preserve all entailments over concept names and role names in  $\Sigma$ .

<sup>5</sup><https://cgi.csc.liv.ac.uk/~konev/software/>

**PROPOSITION 3.3** ([15, 34]). *Let  $\mathcal{T}$  be an  $\mathcal{L}$ -TBox and let  $\Sigma$  be a signature. Additionally, let  $\mathcal{M}_S$  be a semantic module and  $\mathcal{M}_\star$  a star module of  $\mathcal{T}$  w.r.t.  $\Sigma$ . Then for all  $\mathcal{L}$ -inclusions  $\alpha$  with  $\text{sig}(\alpha) \subseteq \Sigma$ , it holds that  $\mathcal{T} \models \alpha$  iff  $\mathcal{M}_S \models \alpha$  and  $\mathcal{T} \models \alpha$  iff  $\mathcal{M}_\star \models \alpha$ .*

In the remainder of the paper, we denote star modules, semantic modules, minimal subsumption modules by  $\mathcal{M}_\star$ ,  $\mathcal{M}_S$  and  $\mathcal{M}_\perp$ .

### 4 UNIFORM INTERPOLATION

Theoretical investigations of uniform interpolation and forgetting for ontologies include [17, 18, 20, 21, 26–29, 31, 45, 46]. Deciding the existence of a uniform interpolant is 2ExpTime-complete for  $\mathcal{ALC}$ -TBoxes, however a uniform interpolant does not always exist in  $\mathcal{EL}$ - and  $\mathcal{ALC}$ -TBoxes [16]. On the other hand, a uniform interpolant always exists for DL-Lite ontologies [18]. Deciding the existence of uniform interpolants of  $\mathcal{EL}$ -ontologies, such as SNOMED CT, is ExpTime-complete [28]. In the worst case, the size of uniform interpolants is triple exponential in the size of the input [31]. Despite the high computational complexity of the problem there are approaches to computing uniform interpolants for light-weight description logics  $\mathcal{EL}$  [27, 28, 31] and DL-Lite ontologies [43]. The system NUI computes  $\mathcal{EL}$ -uniform interpolants for  $\mathcal{EL}$ -ontologies [17]. For description logics extending  $\mathcal{ALC}$  various algorithms for uniform interpolation and forgetting have been developed (either using a resolution-based approach or an Ackermann-based approach) [20, 21, 26, 45, 46]. Performance evaluations of the LETHE<sup>6</sup> and FAME<sup>7</sup> tools have shown good performance on medium size ontologies, both in terms of runtime and success rates [22, 44]. Different to NUI, which computes UIs directly using a generation approach, LETHE computes UIs by forgetting concept and role names not in the input signature. Because the problem is not generally solvable for  $\mathcal{ALC}$ , the UIs are expressed in an extended language using fresh definer symbols to finitely represent infinite UIs, which can arise when there are cyclic dependencies over the forgetting symbols [20, 21]. FAME has been developed to handle large ontologies [44], but the solutions are under-approximations, because they are generally weaker than UIs. NUI computes precise UIs, but target ontologies are limited to  $\mathcal{ELH}^r$ -terminologies.

For ontology extraction we use the notion of uniform interpolation (or deductive forgetting) [17, 29] rather than semantic forgetting [10, 42], because solutions must be in the same language as the input ontology. Tools based on a semantic forgetting approach tend to compute solutions in a more expressive logic, which would not be satisfactory for modellers in the present use case.

*Definition 4.1 (Uniform Interpolation).* Let  $\mathcal{T}$  be an  $\mathcal{L}$ -TBox and let  $\Sigma$  be a signature. A finite set  $\mathcal{U}$  of  $\mathcal{L}$ -inclusions is an  *$\mathcal{L}$ -uniform interpolant (UI)* of  $\mathcal{T}$  for  $\Sigma$  if the following conditions are satisfied: (i)  $\text{sig}(\mathcal{U}) \subseteq \Sigma$ , and (ii) for every  $\mathcal{L}$ -inclusion  $\alpha$  with  $\text{sig}(\alpha) \subseteq \Sigma$ ,  $\mathcal{T} \models \alpha$  iff  $\mathcal{U} \models \alpha$ .

The following example from SNOMED CT illustrates the application of forgetting. To simplify the presentation, we abbreviate concept names as follows:

<sup>6</sup><http://www.cs.man.ac.uk/~koopmanp/lethe/>

<sup>7</sup><http://www.cs.man.ac.uk/~schmidt/sf-fame/>

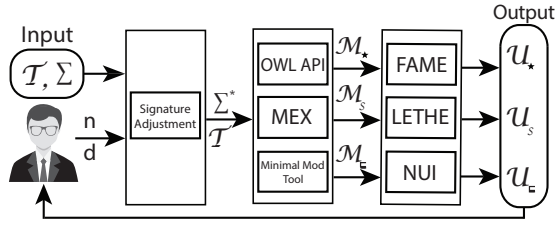


Figure 1: Workflow for computing UIs for the adjustment  $\Sigma^*$  of the signature  $\Sigma$

$A$	Drug_interaction_with_drug (finding)
$A_1$	Drug_interaction (finding)
$A_2$	Drug_or_medicament (substance)
$B$	Substance (substance)
$X$	Adverse_drug_interaction_with_drug (disorder)
$Y$	Adverse_drug_interaction (disorder)
$r$	Associated_with (attribute)

Let  $\mathcal{T} = \{A \sqsubseteq A_1, A_2 \sqsubseteq B, X \sqsubseteq Y \sqcap A \sqcap \exists r.A_2\}$  be a description logic TBox and let  $\Sigma_1 = \{A_1, X\}$  and  $\Sigma_2 = \{A_1, B, X, r\}$  be two sets of terms of interest. Then,  $\mathcal{U} = \{X \sqsubseteq A_1\}$  is a uniform interpolant of  $\mathcal{T}$  for  $\Sigma_1$ , whereas the set containing also the axiom  $X \sqsubseteq \exists r.B$  is a uniform interpolant of  $\mathcal{T}$  for the larger signature  $\Sigma_2$ . The star module of  $\mathcal{T}$  for  $\Sigma_1$  is  $\{A \sqsubseteq A_1, X \sqsubseteq Y \sqcap A \sqcap \exists r.A_2\}$ , which contains concept names  $A, A_2, Y$  and role name  $r$  that are not in  $\Sigma_1$ .

## 5 WORKFLOW

In this section, we present a workflow for forgetting in real-world scenarios, as shown in Figure 1, which includes the following stages: signature adjustment, ontology modularity, forgetting, and evaluation and feedback from a domain expert.

### 5.1 Signature Adjustment

Real-world signatures are typically (a) a selection of a few specific concept names that the user is interested in, and (b) a list of concept names already in use in the domain (a refset), or provided by a domain expert. To account for the different types of signatures, we present two signature adjustment methods: signature extension and signature partition.

*Signature Extension.* Previous evaluations of modularization and forgetting tools on ontologies typically involved the use of random signatures, genuine seed signatures [41], or directly used subsets of ontologies [16]. These signatures do not reflect real-world scenarios of how users or developers would use modules and uniform interpolants of the SNOMED CT ontology.

The evaluation in [7] has shown that minimal subsumption modules for small random signatures are often empty. For instance, a user interested in ‘anticipatory care medication’ or ‘heart structure’ may use a signature consisting of these terms to extract modules from SNOMED CT only to discover that the module is empty. This can also happen for star and semantic modules. Applied to refsets of concept names the forgetting tools yield mere taxonomies, which could be computed more efficiently using the dedicated ELK tool.<sup>8</sup>

<sup>8</sup><https://www.cs.ox.ac.uk/isg/tools/ELK/>

---

### Algorithm 1 Signature-Extension( $\mathcal{T}, \Sigma, n, d$ )

---

**Input:** Normalised Terminology  $\mathcal{T}$ , signature  $\Sigma$ , no. of iterations  $n$ , role depth  $d$

**Output:**  $\Sigma_d^+$

```

1:  $\Sigma_d^+ := N_C \cap \text{sig}(\mathcal{T}) \cap \Sigma$ 
2: while  $n > 0$  do
3:   for  $A \sqsubseteq C \in \mathcal{T}$  with  $A \in \Sigma_d^+$  do
4:     if  $\text{roleDepth}(C) \leq d$  then
5:        $\Sigma_d^+ := \Sigma_d^+ \cup \text{sig}(C)$ 
6:     end if
7:   end for
8:    $n := n - 1$ 
9: end while

```

---



---

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

---

**Input:** Terminology  $\mathcal{T}$ , Signature  $\Sigma$

**Output:** extended signatures  $\Sigma_1^+, \dots, \Sigma_n^+$

```

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

```

---

We have found that more informative UIs can be obtained if the input signature is extended with roles and their target concepts up to a certain nesting from the definitions of input terms. The solution proposed in this paper is to use the axioms of the ontology for enriching the user defined signature in order to obtain more relevant extracts from the ontology. In practice the question arises which axioms to choose and how (and how much) a signature should be extended. Based on discussions with developers from IHTSDO, we propose Algorithm 1 for signature extension. The algorithm uses the function  $\text{roleDepth}(C)$  mapping an  $\mathcal{EL}$ -concept  $C$  to a positive integer which is recursively defined as:

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

In the algorithm, for every axiom  $A \sqsubseteq C$  in  $\mathcal{T}$ , where  $A$  is a concept name in  $\Sigma_d^+$ , the right-hand side  $C$  is included in  $\Sigma_d^+$  if the role depth of  $C$  is at most  $d$ . The maximal role depth of any concept description occurring in SNOMED CT is 2. For ontologies other than SNOMED CT, Algorithm 1 can be adjusted by modifying the iteration number  $n$  and role depth according to domain experts’ suggestions. After evaluation of the extended signature and the UI, further adjustments can be made and the process iterated until a suitable UI is obtained. According to our experiments and discussions with domain experts at IHTSDO, for the SNOMED CT setting  $n = 1$  and  $d = 2$  is sufficient.

*Signature Partition.* Manual inspection is harder in large UIs for large refsets. To obtain a smaller number of closely related concept

and role names, we devised Algorithm 2 performing signature partitioning. The algorithm helps focusing the signature down to a few sets of highly related terms.

Algorithm 2 uses the function `ExtractStarModule( $\Sigma, \mathcal{T}$ )` in Line 1, which is provided by the OWL API<sup>9</sup> for computing star modules. Even though the function accepts general TBoxes  $\mathcal{T}$  formulated in OWL 2, we only use it as a convenient way to compute the star module  $\mathcal{M}_\star$  of SNOMED CT. The function `Classify( $\mathcal{M}_\star$ )` in Line 2 then calls the ELK reasoner to classify  $\mathcal{M}_\star$ . Subsequently, we reduce the computed class hierarchy to the symbols in the input signature. We obtain a classification  $\mathcal{H}'$  of the concept names in the input signature. The method `Partition( $\mathcal{H}'$ )` in Line 3 separates the class hierarchy into several unrelated parts. Viewing a class hierarchy as a directed graph, this can be implemented by computing the connected components of the graph. 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  $n$  disjoint sub-hierarchies:  $\mathcal{H}_1, \dots, \mathcal{H}_n$ . The loop from Line 4-6, first computes the signature  $\Sigma_i$  of hierarchy  $\mathcal{H}'_i$  and then extends  $\Sigma_i$  to  $\Sigma_i^+$  using the function `Signature-Extension( $\mathcal{M}_\star, \Sigma_i$ )` presented in Algorithm 1.

## 5.2 Modularity Meets Forgetting

In this section, we first compare the notions of locality-based modules [12], semantic modules [16] and minimal subsumption modules [6, 7]. We then show the correctness of our workflow, i.e., doing forgetting on these modules is logically equivalent to doing forgetting on the original ontology.

The following example illustrates the difference between the three module notions. To simplify the presentation, we use the concept names  $A, A_1, A_2, B, X$  and  $Y$  to abbreviate SNOMED CT concept names as follows:

$A$	Mesoblastic_nephroma
$A_1$	Neoplasm_uncertain_whether_benign_or_malignant
$A_2$	Complex_mixed_AND/OR_stromal_neoplasm
$B$	Neoplasm
$X$	Neoplasm_and/or_hamartoma
$Y$	Tumor

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 minimal subsumption modules of  $\mathcal{T}$  w.r.t.  $\Sigma$ :  $\{\alpha_1, \alpha_2\}$  and  $\{\alpha_1, \alpha_3\}$ . Neither  $\{\alpha_1, \alpha_2\}$  nor  $\{\alpha_1, \alpha_3\}$  is sufficient to preserve the entailment  $A \sqsubseteq B$  that only uses symbols in  $\Sigma$ . The semantic module and star module of  $\mathcal{T}$  w.r.t.  $\Sigma$  are each  $\{\alpha_1, \alpha_2, \alpha_3\}$ .

For  $\mathcal{T}' = \{\alpha_1, \alpha_2, \alpha_3, \alpha_5\}$  with  $\alpha_5 := B \equiv Y$ ,<sup>10</sup> the star module w.r.t.  $\Sigma$  is  $\mathcal{T}'$  itself. However, the minimal subsumption modules and semantic module of  $\mathcal{T}'$  w.r.t.  $\Sigma$  coincide with corresponding those of  $\mathcal{T}$ , respectively. (End of example)

The minimal depleting module of an  $\mathcal{ELI}$ -terminology w.r.t.  $\Sigma$  is always a subset of the respective star module w.r.t.  $\Sigma$  [16]. A star

<sup>9</sup><http://owlapi.sourceforge.net/>

<sup>10</sup>Neoplasm ( $B$ ) and Tumor ( $Y$ ) are treated as synonyms in SNOMED CT and share the same identifier. In order to illustrate the difference between semantic modules and star module, we add  $\alpha_5$  in this example.

module coincides with a minimal depleting module when the terminology contains no concept definitions (cf. Proposition 38 in [16]). Different to the notion of semantic modules, which is defined in terms of a model-theoretic inseparability relation, the notion of a subsumption module is defined in terms of entailment queries. It was shown that a minimal subsumption module is contained in the respective semantic module [2].

Both semantic modules and star modules, each yield a unique subset of a given TBox w.r.t. a signature. On the other hand, there may exist several, up to exponentially many, minimal subsumption modules of  $\mathcal{T}$  for a signature (cf. Example 6 in [7]).

As computing uniform interpolants is a difficult task especially for large ontologies, the size and complexity of the input ontology directly influences the computation time. Instead of computing uniform interpolants on the whole ontology, the idea we follow in this paper is that speed up can be achieved by computing uniform interpolants from ontology modules. The required correctness guarantee is provided by:

**PROPOSITION 5.1.** *Let  $\mathcal{T}$  be an  $\mathcal{L}$ -TBox and let  $\Sigma$  be a signature. If  $\mathcal{U}$  is a uniform interpolant of  $\mathcal{M}_\sqsubseteq$  ( $\mathcal{M}_S$  or  $\mathcal{M}_\star$ ) for  $\Sigma$ , then  $\mathcal{U}$  is a uniform interpolant of  $\mathcal{T}$  for  $\Sigma$ .*

**PROOF.** Let  $\mathcal{U}$  be a uniform interpolant of  $\mathcal{M}_\sqsubseteq$  ( $\mathcal{M}_S$  or  $\mathcal{M}_\star$ ) for  $\Sigma$ . Then  $\text{sig}(\mathcal{U}) \subseteq \Sigma$  by Definition 4.1. To show that, in each case,  $\mathcal{U}$  is a uniform interpolant of  $\mathcal{T}$ , we show that  $\mathcal{U}$  preserves the  $\Sigma$ -entailments of  $\mathcal{T}$ . Let  $\alpha$  be an  $\mathcal{L}$ -inclusion with  $\text{sig}(\alpha) \subseteq \Sigma$ . We show that the following are equivalent: (i)  $\mathcal{U} \models \alpha$ ; (ii)  $\mathcal{M} \models \alpha$ ; and (iii)  $\mathcal{T} \models \alpha$ , where  $\mathcal{M}$  ranges over  $\mathcal{M}_\sqsubseteq$ ,  $\mathcal{M}_S$  and  $\mathcal{M}_\star$ . The equivalence of (i) and (ii) holds by Definition 4.1, and the equivalence of (ii) and (iii) holds by Definition 3.2 for  $\mathcal{M} = \mathcal{M}_\sqsubseteq$ , and it follows from Proposition 3.3 for  $\mathcal{M} = \mathcal{M}_S$  and  $\mathcal{M} = \mathcal{M}_\star$ .  $\square$

The use of star modularization has already been explored in the implementation of LETHE.

## 6 EVALUATION

The goal of the evaluation was to measure the performance of the workflow in terms of success rate and computation time for the different modularization and forgetting tools. We used the OWL API, the MEX tool and the tool in [7] to compute star modules, semantic modules and minimal subsumption modules. To perform forgetting we used NUI, FAME and LETHE. The experiments with FAME and LETHE were conducted on machines equipped with Intel Xeon CPU E5-2640 v3 running at 2.60GHz with 32GB of RAM. As NUI ran only on 32 bit architectures, its experiments were run on machines equipped with Intel Xeon 4 Duo CPU at 2.50 GHz and 64GiB of RAM. The timeout was set to 1 hour.<sup>11</sup> We were interested in preserving  $\mathcal{EL}$ -inclusions for SNOMED CT and NCIT.

Note that in interpreting the results of the tables one should take into account that NUI computes UIs which are formulated in  $\mathcal{EL}$ , whereas the target logic of LETHE and FAME is  $\mathcal{ALC}$ . The results are therefore not directly comparable.

<sup>11</sup>If a forgetting tool terminated normally within the timeout, we counted this as a successful run. Our experiments showed that success rates barely increased once the timeout increased to more than 1 hour. We therefore limited the timeout to 1 hour.

Tool		FAME						LETHE						NUI					
$ \Sigma^* \cap \text{NC} $	No. of Sig	Success Rate (%)			Med. Time of Forgetting			Success Rate (%)			Med. Time of Forgetting			Success Rate (%)			Med. Time of Forgetting		
		$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$
[0, 150]	110	75.45	89.09	94.54	0.28	0.28	0.27	92.73	100.00	100.00	1.29	1.11	0.96	97.27	99.09	100.00	0.01	0.01	0.01
[150, 300]	13	38.46	46.15	76.92	3 600.00	3 600.00	13.51	61.54	84.62	92.31	9.72	4.10	3.22	100.00	100.00	100.00	0.09	0.04	0.02
[300, 116 865]	42	4.76	11.90	14.28	3 600.00	3 600.00	3 600.00	14.29	26.20	28.61	3 600.00	3 600.00	3 600.00	61.90	78.57	84.62	59.05	20.84	16.10

**Table 1: Success rate and median computation time (s) of computing UIs using NHS refsets as signatures by FAME, LETHE and NUI on SNOMED CT**

Tool	FAME			LETHE			NUI		
$\mathcal{T}$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$
Min.	0.10	0.14	0.16	1.45	0.61	0.59	0.03	0.02	0.02
Max.	3 600.00	3 600.00	3 600.00	3 600.00	19.23	1.79	0.14	0.08	0.04
Avg.	2 880.04	960.21	480.23	1 870.11	2.50	0.99	0.05	0.04	0.03
Med.	3 600.00	0.22	0.20	2 194.69	1.24	0.88	0.05	0.04	0.03
Succ.(%)	21.42	78.57	92.86	50.00	100.00	100.00	100.00	100.00	100.00

**Table 2: Computation time (s) of computing UIs from different modules of SNOMED CT by FAME, LETHE and NUI using 14 signatures obtained as extensions of the ERA refset**

Tool	FAME			LETHE			NUI		
$\mathcal{T}$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$
Min.	0.12	0.13	0.13	0.62	0.61	0.56	0.03	0.01	0.02
Max.	3 600.00	3 600.00	0.98	31.68	24.39	1.97	0.14	0.08	0.04
Avg.	527.07	88.07	0.30	4.24	1.68	0.86	0.05	0.03	0.03
Med.	0.33	0.21	0.25	1.59	0.74	0.85	0.05	0.03	0.03
Succ.(%)	85.37	97.56	100.00	100.00	100.00	100.00	100.00	100.00	100.00

**Table 3: Computation time (s) of UIs on different modules of NCIt by FAME, LETHE and NUI**

Tool	FAME			LETHE			NUI		
$\mathcal{T}$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$	$M_\star$	$M_S$	$M_\square$
Min.	0	0	0	0	0	0	0	0	0
Max.	210	231	232	95	95	93	101	101	103
Avg.	32.94	35.77	48.65	18.46	15.70	18.26	20.18	20.18	22.98
Med.	25.00	23.00	28.00	12.00	10.00	12.00	13.00	13.00	14.50

**Table 4: Sizes of UIs from different modules by FAME, LETHE and NUI on the NCIt ontology**

## 6.1 SNOMED CT

The evaluation was performed on the  $\mathcal{EL}$  fragment of SNOMED CT (version Jan 2016), obtained by removing 129 axioms not in  $\mathcal{EL}$  and 20 GCI axioms of the form  $C \sqsubseteq A$ . We used two sets of real-world signatures, *NHS refsets*<sup>12</sup> and *ERA refsets*, as inputs.

*NHS refsets.* The NHS refsets were provided by the National Health Service (NHS) in the UK. Their purpose is described in <http://bit.ly/2k7AtsK>. After discussion with domain experts, we performed the experiments by using extended signatures ( $\Sigma_2^+$ ,  $n = 1$ ,  $d = 2$ ). In total, we obtained 165 signatures of concept names all contained in the SNOMED CT ontology. The signatures consisted of 2–116 865 concept names and 0–33 role names. We separated the signatures into three different groups according to the number of concept names they contained. For each signature we computed the three types of modules, their precision rate and the UIs for each forgetting tool. As definition of the precision rate of a module  $\mathcal{M}$  w.r.t. a signature  $\Sigma$  we used:

$$\text{Precision}(\mathcal{M}, \Sigma) := |(\Sigma \cap \text{sig}(\mathcal{M})) \cup \{\top\}| / |\text{sig}(\mathcal{M})|,$$

which is a slight adaptation from [25] with  $\top$  added in the numerator since every individual is an instance of the  $\top$  concept.

The results show that the average precision rate of the star modules, semantic modules and minimal subsumption modules w.r.t. the signatures were around 72%, 71% and 78%, respectively. In contrast, the precision rate of UIs is 100% (cf. Definition 4.1) unless definers occur in the result. In the case of LETHE the results for seven UIs obtained from star modules contained definer symbols. This was also the case for one of the results computed for the

semantic and minimal subsumption module. There were no such cases for the ERA refset and NCIt evaluation.

As shown in Table 1, for computing UIs on small signatures containing less than 150 concept names, running the forgetting tools on precomputed minimal subsumption modules made it possible to compute UIs within 1 hour with a success rate<sup>13</sup> of 100% for LETHE and NUI. Precomputing semantic modules is also competitive for such small signatures. The success rates dropped rapidly with increasing signature size due to quickly growing ontology modules for signatures containing more than 300 concept names. This effect can be observed for FAME and LETHE, and to a lesser extend for NUI. The median times of forgetting reported in this section are median values for all three types of modules and do not include the time for extracting modules. If forgetting was not completed within the timeout, the forgetting time of this run was taken to be 1 hour. Where the success rate was less than 50% the median computation time was therefore taken to be 1 hour. We see it took less time for all tools to forget symbols from minimal subsumption modules than from semantic modules and star modules.

*ERA refset.* For the second part of the evaluation, we used the European Renal Association (ERA) refset of symbols from SNOMED CT, which has been provided by IHTSDO. The ERA refset contains a list of primary renal diseases, designed for use in renal centres and registries [40]. We first obtained 14 small disjoint signatures by partitioning the ERA refset. Next, we applied Algorithm 1 to extend these signatures in order to relate symbols in the ERA refset with symbols representing diseases, body structure, role names, etc. The resulting signatures consisted of 5–40 concept names and 0–8 role names. For each of the 14 signatures, we computed the three different modules of SNOMED CT. These modules together with their respective producing signatures were then taken as input for the systems FAME, LETHE and NUI to compute UIs. The times to compute UIs are summarized in Table 2 in terms of the minimal, maximal, average, and median time for all runs, including successful and unsuccessful cases. The basis for the success rate was the successful cases that finished within a timeout of 1 hour. It becomes evident in

<sup>12</sup><https://isd.digital.nhs.uk/trud3/user/guest/group/0/pack/40>

<sup>13</sup>If a tool terminated within 1 hour, we counted it as a successful run.

	#sig <sub>NC</sub> (M <sub>★</sub> )	#sig <sub>NC</sub> (M <sub>S</sub> )	#sig <sub>NC</sub> (M <sub>⊖</sub> )
Min.	7	6	6
Max.	319	172	95
Avg.	97.9	47.1	24.4
Med.	96.0	41.0	14.5

**Table 5:** No. of concept names in different modules of SNOMED CT

	#sig <sub>NR</sub> (M <sub>★</sub> )	#sig <sub>NR</sub> (M <sub>S</sub> )	#sig <sub>NR</sub> (M <sub>⊖</sub> )
Min.	0	0	0
Max.	17	15	8
Avg.	5.0	4.0	3.5
Med.	4.0	3.5	3.5

**Table 6:** No. of role names in different modules of SNOMED CT

	#M <sub>★</sub>	#M <sub>S</sub>	#M <sub>⊖</sub>
Min.	6	2	2
Max.	304	158	64
Avg.	92.7	41.8	14.8
Med.	91.0	38.0	6.5

**Table 7:** No. of axioms in different modules of SNOMED CT

	#U <sub>★</sub>	#U <sub>S</sub>	#U <sub>⊖</sub>
Min.	(2)	2	2
Max.	(20)	40	42
Avg.	(8.71)	10.43	10.14
Med.	(7.0)	7.5	6.5

**Table 8:** No. of axioms in UIs of LETHE for modules in Table 7

Table 2 that precomputing semantic modules and minimal subsumption modules considerably reduced the computation time and, thus, increased the success rate of computing UIs for LETHE and NUI to 100%. In particular in the case of minimal subsumption modules, the UI for any signature could be computed within 1.79 seconds by LETHE for all signatures. Contrast this with the fact that the success rate was only 21.42% for FAME when combined with star modules.

We need to keep in mind however that computing minimal subsumption modules is computationally more expensive. The time needed to compute the minimal subsumption modules for the 14 signatures ranged from 1 to 939 seconds, whereas semantic modules and star modules were computed in less than 5 seconds.

The success rate of computing UIs from star modules is at most 50% for FAME and LETHE. Using semantic modules instead increases the success rate to 100% for LETHE, but only to about 79% for FAME, cf. last row of Table 2. Hence, depending on the timeout, the use of minimal subsumption modules can enable the computation of UIs.

## 6.2 NCIt Ontology

We also undertook an evaluation on the acyclic  $\mathcal{EL}$  fragment of NCIt ontology containing around 150 000 axioms. The real-world signatures used were the sets of drugs approved by the Food and Drug Administration (FDA) for 41 types of cancers, consisting of 2–162 concept names and 0–17 role names. The experimental setup was similar to that for SNOMED CT.

The computation times of forgetting are shown in Table 3. For star modules, semantic modules and minimal subsumption modules the average precision rates were around 49%, 62% and 85%. As the NCIt ontology was 50% smaller and less complicated than SNOMED CT (the latter contains nested existential restriction) in most cases the success rates were higher than in the experiments in Section 6.1. In general, computing UIs was faster on minimal subsumption modules, especially for LETHE and FAME. Table 4 shows the size of UIs that were computed by the three tools applied to the three types of modules of NCIt. They were almost the same for LETHE and NUI, while the results were about 50% larger for FAME. LETHE performs denormalisation, making UIs more user-friendly.

## 7 DISCUSSION

It was found that neither NUI, FAME nor LETHE could compute UIs on SNOMED CT and NCIt ontologies without precomputing ontology modules in real-world signatures with less than 35% of the symbols in the ontology. In most unsuccessful cases, FAME and LETHE did not terminate within a reasonable amount of time, whereas NUI ran out of memory. The results in Section 6 show that precomputing minimal subsumption and semantic modules can considerably speed up the process of computing ontology extracts, especially for the tools FAME and LETHE. The results, when using the workflow, suggest forgetting on a semantic module should be tried

first as it was the fastest configuration for cases where forgetting succeeds. Otherwise, forgetting on a minimal subsumption module could be tried, with which forgetting may be successful but which may take longer to extract.

In the rest of the section, we analyse the reasons why module extraction techniques can help to optimize forgetting tools by analyzing the detailed statistics (Table 5–8) of the ERA reset experiments of computing UIs on SNOMED CT.

*Smaller Module.* Table 7 shows that, on average, the size (number of axioms) of minimal subsumption modules were more than 2 times smaller than semantic modules, and 5 times smaller than star modules (even 13 times smaller than star modules according to median value). Thus more precise, smaller modules make forgetting easier.

*Fewer Symbols to Forget.* As can be seen in Table 5, the number of concept names that occur in minimal subsumption modules was 53% and 25% of those in semantic modules and star modules on average. As the interpolation signature was the same for all modules, forgetting on subsumption modules has a lot fewer concept names to forget, and similarly for semantic modules. Although forgetting role names is more difficult than forgetting concept names, the number of role names did not vary much on average for the different modules, cf. Table 6.

*Special Role ‘RoleGroup’.* In SNOMED CT, a special role name, called ‘RoleGroup’, maintains correct inferences and semantic meaning for complex concept expressions related to, e.g., multiple sites and morphologies [38]. The presence of ‘RoleGroup’ comes with nested existential expression. We found that nested existential expressions made forgetting harder. However, inspection has revealed that semantic modules, especially minimal subsumption modules, had considerably fewer axioms containing ‘RoleGroup’.

*Size of UIs.* Table 8 shows the sizes of LETHE’s UIs<sup>14</sup> in the experiment with ERA reset. The values in brackets are computed only over successful cases. The UIs for semantic and minimal subsumption modules were similar in size. Comparing Tables 7 and 8, in this experiment the UI sizes were close to the sizes of minimal subsumption modules and considerably smaller than semantic modules. This is much better than the worst-case upper bound suggests. However, very large UIs were obtained in the experiments with NHS resets where the signatures were much larger. This aspect deserves further investigation.

## 8 CONCLUSION AND FUTURE WORK

Ontology development for applications such as clinical data analytics needs to be facilitated by automated tools. The goal is to create ontology abstractions of large ontologies that only involve the terms of interest to the developer or end-user. Despite the fact

<sup>14</sup>The size of a UI was the number of axioms contained in the UI.

that ontology modules tend to exhibit low precision rates and current forgetting tools have difficulties when applied directly to large ontologies for very small signatures, we show the two approaches can be successfully combined. Our method provides a feasible approach with high precision to compute uniform interpolants for realistic, small-sized signatures of prominent, large ontologies. In future work, we expect to further explore other signature adjustment algorithms, and evaluate the quality of the obtained modules and uniform interpolants with domain experts. To make better use of current module extraction and uniform interpolation techniques in real-world situation, we will update current module extraction and forgetting techniques according to feedback from domain experts. Besides, the algorithms for computing minimal subsumption modules and semantic modules are expected to be updated in order to deal with more expressive ontologies.

*Acknowledgements.* We thank the reviewers for very useful comments, Yizheng Zhao and Patrick Koopmann for their help with the systems FAME and LETHE, and Boris Konev for providing the NUI system. The research was undertaken in IAA 228 Project funded by the EPSRC and IHTSDO. The first author was partially supported by the SIRIUS Centre, Oslo. Initial findings appear in [3].

## REFERENCES

- [1] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic*. Cambridge University Press.
- [2] Jieying Chen. 2018. *Knowledge Extraction from Description Logic Terminologies*. Ph.D. Dissertation. University of Paris-Saclay, France.
- [3] Jieying Chen, Ghadah Alghamdi, Renate A. Schmidt, Dirk Walther, and Yongsheng Gao. 2019. Modularity Meets Forgetting: A Case Study with the SNOMED CT Ontology. In *Proc. of DL'19*, Vol. 2373. CEUR-WS.org.
- [4] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. 2017. Zooming in on Ontologies: Minimal Modules and Best Excerpts. In *Proc. of ISWC'17*. 173–189.
- [5] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. 2019. Computing Minimal Projection Modules for  $ELH'$ -Terminologies. In *Proc. of JELIA'19*. 355–370.
- [6] Jieying Chen, Michel Ludwig, and Dirk Walther. 2016. On Computing Minimal  $\mathcal{E}\mathcal{L}$ -Subsumption Modules. In *Proc. of WOMoCoE'16*.
- [7] Jieying Chen, Michel Ludwig, and Dirk Walther. 2018. Computing Minimal Subsumption Modules of Ontologies. In *Proc. of GCAI'18*. 41–53.
- [8] Paul Doran, Valentina A. M. Tamma, Terry R. Payne, and Ignazio Palmisano. 2009. An entropy inspired measure for evaluating ontology modularization. In *Proc. of K-CAP'09*. 73–80.
- [9] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, and K. Wang. 2006. Forgetting in Managing Rules and Ontologies. In *Proc. of WT'06*. 411–419.
- [10] Thomas Eiter and Kewen Wang. 2008. Semantic forgetting in answer set programming. *Artificial Intelligence* 172, 14 (2008), 1644–1672.
- [11] Jennifer Golbeck, Gilberto Fragoso, Frank W. Hartel, James A. Hendler, Jim Oberthaler, and Bijan Parsia. 2003. The National Cancer Institute's Thésaurus and Ontology. *J. Web Semant.* 1 (2003), 75–80.
- [12] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. 2008. Modular Reuse of Ontologies: Theory and Practice. *J. Artif. Intell. Res.* 31, 1 (2008), 273–318.
- [13] Pascal Hitzler and Cogan Shimizu. 2018. Modular Ontologies as a Bridge Between Human Conceptualization and Data. In *Proc. of ICCS'18*. 3–6.
- [14] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga Llavori. 2008. Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support. In *Proc. of ESWC'08*. 185–199.
- [15] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2009. *Formal Properties of Modularisation*. Springer Verlag, Berlin, Heidelberg, 25–66.
- [16] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.* 203 (2013), 66–103.
- [17] Boris Konev, Dirk Walther, and Frank Wolter. 2009. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *Proc. of IJCAI'09*. 830–835.
- [18] Roman Kontchakov, Frank Wolter, and Michael Zakharyashev. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.* 174, 15 (2010), 1093–1141.
- [19] Patrick Koopmann and Jieying Chen. 2017. Computing ALCH-Subsumption Modules Using Uniform Interpolation. In *Proc. of SOQE'17*. 51–66.
- [20] Patrick Koopmann and Renate A. Schmidt. 2013. Forgetting Concept and Role Symbols in  $\mathcal{ALCH}$ -Ontologies. In *Proc. of LPAR'13*, Vol. 8312. 552–567.
- [21] Patrick Koopmann and Renate A. Schmidt. 2013. Uniform Interpolation of  $\mathcal{ALC}$ -Ontologies Using Fixpoints. In *Proc. of FroCoS'13*, Vol. 8152. 87–102.
- [22] Patrick Koopmann and Renate A. Schmidt. 2015. LETHE: A Saturation-Based Tool for Non-Classical Reasoning. In *Proc. of ORE'15*.
- [23] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. 2003. Propositional Independence: Formula-Variable Independence and Forgetting. *J. Artif. Intell. Res.* 18 (2003), 391–443.
- [24] Fangzhen Lin and Ray Reiter. 1994. Forget It!. In *Proceedings of the AAAI Fall Symposium on Relevance*. 154–159.
- [25] Pablo López-García, Martin Boeker, Arantza Illarramendi, and Stefan Schulz. 2012. Usability-driven pruning of large ontologies: the case of SNOMED CT. *Journal of the American Medical Informatics Association* 19, e1 (2012).
- [26] Michel Ludwig and Boris Konev. 2014. Practical Uniform Interpolation and Forgetting for ALC TBoxes with Applications to Logical Difference. In *Proc. of KR'14*.
- [27] Michel Ludwig and Dirk Walther. 2016. Towards a Practical Decision Procedure for Uniform Interpolants of  $\mathcal{E}\mathcal{L}$ -TBoxes – a Proof-Theoretic Approach. In *Proc. of GCAI'16*. 147–160.
- [28] Carsten Lutz, İnanç Seylan, and Frank Wolter. 2012. An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic EL. In *Proc. of KR'12*.
- [29] Carsten Lutz and Frank Wolter. 2011. Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In *Proc. of IJCAI'11*. 989–995.
- [30] Till Mossakowski, Mihai Codrescu, Fabian Neuhaus, and Oliver Kutz. 2015. The Distributed Ontology, Modeling and Specification Language – DOL. In *The Road to Universal Logic*, Arnold Koslow and Arthur Buchsbaum (Eds.). Vol. 2. Birkhäuser, 489–520.
- [31] Nadeschda Nikitina and Sebastian Rudolph. 2014. (Non-)Succinctness of uniform interpolants of general terminologies in the description logic EL. *Artif. Intell.* 215 (2014), 120–140.
- [32] Riku Nortje, Katarina Britz, and Thomas Meyer. 2013. Reachability Modules for the Description Logic SRIQ. In *Proc. of LPAR'13*. 636–652.
- [33] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. 2016. Module Extraction in Expressive Ontology Languages via Datalog Reasoning. *J. Artif. Intell. Res.* 55 (2016), 499–564.
- [34] U. Sattler, T. Schneider, and M. Zakharyashev. 2009. Which Kind of Module Should I Extract?. In *Proc. of DL'09 (CEUR Workshop Proceedings)*, Vol. 477. CEUR-WS.org.
- [35] Stefan Schlobach. 2004. Explaining Subsumption by Optimal Interpolation. In *Proc. of JELIA'19*. 413–425.
- [36] Kent Spackman. 2000. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with Snomed-RT. *Journal of the American Medical Informatics Association* (01 2000).
- [37] Kent Spackman, Keith Campbell, and RA Cote. 1997. SNOMED RT: a reference terminology for health care. In *Proc. of AMIA'97*.
- [38] Kent A. Spackman, Robert Dionne, Eric Mays, and Jason Weis. 2002. Role grouping as an extension to the description logic of ontology, motivated by concept modeling in SNOMED. In *Proc. of AMIA'02*.
- [39] Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra. 2009. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. Lecture Notes in Computer Science, Vol. 5445. Springer Verlag.
- [40] Gopalakrishnan Venkat-Raman, Charles R.V. Tomson, Yongsheng Gao, Ronald Cornet, Benedicte Stengel, Chris Reid Carola Gronhagen-Riska, Christian Jacquelinet, Elke Schaeffner, Els Boeschoten, Francesco Casino, Frederic Collart, Johan De Meester, Oscar Zurriaga, Reinhard Kramer, Kitty J. Jager, and Keith Simpson. 2012. New primary renal diagnosis codes for the ERA-EDTA. *Nephrology Dialysis Transplantation* 27, 12 (2012), 4414–4419.
- [41] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider, and Dmitry Tsarkov. 2013. Empirical Study of Logic-Based Modules: Cheap Is Cheerful. In *Proc. of DL'13*. 144–155.
- [42] Kewen Wang, Zhe Wang, Rodney Topor, Jeff Z. Pan, and Grigoris Antoniou. 2014. Eliminating Concepts and Roles from Ontologies in Expressive Description Logics. *Computational Intelligence* 30, 2 (2014), 205–232.
- [43] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. 2010. Forgetting for knowledge bases in DL-Lite. *Ann. Math. Artif. Intell.* 58, 1-2 (2010), 117–151.
- [44] Yizheng Zhao, Ghadah Alghamdi, Renate A. Schmidt, Hao Feng, Giorgos Stoilos, Damir Juric, and Mohammad Khodadadi. 2019. Tracking Logical Difference in Large-Scale Ontologies: A Forgetting-Based Approach. In *Proc. of AAAI'19*.
- [45] Yizheng Zhao and Renate A. Schmidt. 2016. Forgetting Concept and Role Symbols in  $\mathcal{ALCOI}\mathcal{H}\mu^*$  ( $\nabla, \Gamma$ )-Ontologies. In *Proc. of IJCAI'16*. 1345–1352.
- [46] Yizheng Zhao and Renate A. Schmidt. 2018. On Concept Forgetting in Description Logics with Qualified Number Restrictions. In *Proc. of IJCAI'18*. 1984–1990.