Concrete Names for Complex Expressions in Ontologies: A Survey of Biomedical Ontologies

Christian Kindermann[1], Martin Georg Skjæveland[2]

[1]Stanford University, 450 Serra Mall, Stanford, USA
[2]University of Oslo, Problemveien 7, 0315 Oslo, Norway

Abstract

The representation of an entity in an ontology may require complex expressions to capture all of its relevant characteristics. If an entity can be *defined* based on its characteristics, then its definition can be explicitly stated in most knowledge representation languages, such as the Web Ontology Language (OWL). Specifically, a domain-specific entity can be identified by a name in an ontology, which can be declared to be logically equivalent to a more complex expression This not only fixes the meaning of the entity in the ontology but also allows its name to replace the more complex expression throughout the ontology. Consistently using concise and informative names for domain-specific entities in an ontology can arguably enhance ontology comprehension, maintenance, and usability in practice. This raises the question of the extent to which entities represented in ontologies are associated with concrete names and how such names are used.

In this paper, we analyze how often named classes in OWL ontologies are defined as logically equivalent to complex expressions. We investigate whether such named classes are consistently used whenever possible and whether they are associated with labels intended for human understanding. Our findings indicate that complex class expressions are frequently declared to be equivalent to named classes in ontologies, and that such named classes are linked to human readable labels. While there seems to be a tendency to encourage the reuse of these names, we also observe a notable number of instances where such named classes are not consistently reused despite being defined.

Keywords

Ontology Engineering, Biomedical Ontology, Web Ontology Language, OWL

1. Introduction

The representation of an entity in an ontology typically involves statements about the entity's characteristics. When an entity can be defined based on its characteristics, the definition may include an informative name by which the entity can be referred to. Specifically, an entity's name may be used instead of its more complex definitional description. Despite the potential benefits of consistently using concise and informative names whenever possible, it has been observed that this practice is not always followed in published ontologies. To illustrate this, we revisit a concrete example taken from the Galen ontology, which was originally presented by Nikitina and Koopmann [1]. Here, the medical concept Clotting is represented as follows:

Proceedings of the International Conference on Biomedical Ontologies 2023, August 28th-September 1st, 2023, Brasilia, Brazil
EMAIL: christian.kindermann@stanford.edu (C. Kindermann); martige@ifi.uio.no (M. G. Skjæveland)
ORCID: 0000-0001-7818-3466 (C. Kindermann); 0000-0002-9736-8316 (M. G. Skjæveland)

$$\begin{aligned}\mathsf{Clotting} \equiv\ & \exists\, \mathsf{actsSpecificallyOn}.(\mathsf{Blood} \\ &\quad \sqcap \exists\, \mathsf{hasPhysicalState}.(\mathsf{PhysicalState} \sqcap \exists\, \mathsf{hasState}.\mathsf{Liquid})) \\ & \sqcap \exists\, \mathsf{hasOutcome}.\mathsf{SolidBlood}\end{aligned}$$

This axiom is arguably complex due to both its size and the nesting of expressions. However, Galen also contains the following axioms:

$$\begin{aligned}\mathsf{LiquidBlood} &\equiv \mathsf{Blood} \sqcap \exists\, \mathsf{hasPhysicalState}.\mathsf{LiquidState} \\ \mathsf{LiquidState} &\equiv \mathsf{PhysicalState} \sqcap \exists\, \mathsf{hasState}.\mathsf{Liquid}\end{aligned}$$

Given these equivalences, the named concept LiquidBlood can be used to simplify the representation of Clotting to

$$\begin{aligned}\mathsf{Clotting} \equiv\ & \exists\, \mathsf{actsSpecificallyOn}.\mathsf{LiquidBlood} \\ & \sqcap \exists\, \mathsf{hasOutcome}.\mathsf{SolidBlood}\end{aligned}$$

The latter representation of Clotting is arguably easier to read, comprehend, and maintain. This observation raises questions about *the frequency of defining* concrete names for complex expressions, *the consistency of using* such names throughout an ontology, and to what extent the use of names *simplifies* the definition of more complex concepts. The contributions presented in this paper are as follows: (i) we propose an approach for identifying named classes with logical definitions in ontologies, (ii) we develop techniques for quantifying the use and lack of reuse of such named classes, and (iii) we use these techniques to conduct an empirical investigation on a large and complex corpus of ontologies in the biomedical domain to shed light on the use of such names in real-world ontologies.

2. Preliminaries

We assume the reader to be familiar with OWL [2] and only fix some terminology. Let N_C, N_I, and N_P be sets of *class names*, *individual names*, and *property names*. A *class* is either a class name or a *complex class* built using OWL class constructors. We will use $\top$ and $\bot$ to denote `owl:Thing` and `owl:Nothing` respectively. We use both OWL Functional Style Syntax [3] and Manchester Syntax [4] to write OWL axioms. An ontology is a set of axioms and we write $\mathcal{O} \models \alpha$ to denote that the ontology $\mathcal{O}$ entails the axiom α. An axiom α is *explicit* in $\mathcal{O}$ if $\alpha \in \mathcal{O}$, and *implicit* if $\alpha \notin \mathcal{O}$ but $\mathcal{O} \models \alpha$. An OWL expression e *occurs* in $\mathcal{O}$ if e is used as a subexpression within an explicit axiom in $\mathcal{O}$.

3. Abbreviations in Ontologies

The Oxford English Dictionary defines the word *abbreviation* to denote "[t]he result of shortening something; an abbreviated or condensed form, esp. of a text; a summary, an abridgement" [5]. So, we define an abbreviation for a complex OWL expression in terms of an equivalent named class. More formally, let A be a named class and C be a complex class expression. Then A is an

$\alpha_1 = \mathsf{SpicyPizza}$ EquivalentTo Pizza and
(hasTopping some (PizzaTopping and (hasSpiciness some Hot)))
$\alpha_2 = \mathsf{SpicyTopping}$ EquivalentTo PizzaTopping and (hasSpiciness some Hot)
$\alpha_3 = \mathsf{SpicyTopping}$ EquivalentTo HotTopping
$\alpha_4 = \mathsf{DiavolaPizza}$ SubClassOf SpicyPizza
$\alpha_5 = \mathsf{DiavolaPizza}$ SubClassOf Pizza and hasCountryOfOrigin value Italy
$\alpha_6 = \mathsf{NapoletanaPizza}$ SubClassOf Pizza and hasCountryOfOrigin value Italy

Figure 1: Example of abbreviations and synonyms in a sample ontology. The named class SpicyPizza is an abbreviation. The classes SpicyTopping and HotTopping are synonyms.

abbreviation for C in an ontology $\mathcal{O}$, if $\mathcal{O} \models \mathit{EquivalentClasses}(\mathsf{A}, \mathsf{C})$. We will refer to the $\mathit{EquivalentClasses}$ axiom as the **definition** of the abbreviation A.

A complex OWL expression can be equivalent to more than just one named class. We refer to equivalent named classes as *synonyms*.[1] In particular, a **synonym** for a named class N in an ontology $\mathcal{O}$ is a named class S s.t. $\mathcal{O} \models \mathit{EquivalentClasses}(\mathsf{S}, \mathsf{N})$ and we will refer to the $\mathit{EquivalentClasses}$ axiom as the synonym's definition. Please note that synonyms are not necessarily abbreviations. However, a synonym for an abbreviation is also an abbreviation (due to transitivity of $\mathit{EquivalentClasses}$).

Both abbreviations and synonyms are notions based on entailment, i.e., an *EquivalentClasses* axiom with exactly two arguments. However, OWL specifies *EquivalentClasses* as an n-ary constructor. So, for the purpose of analyzing how abbreviations and synonyms are specified in ontologies, we introduce the notion **syntactic definitions types** for both abbreviations and synonyms. In particular, an axiom of the form $\mathit{EquivalentClasses}(\mathsf{A}, \mathsf{C})$ and $\mathit{EquivalentClasses}(\mathsf{S}, \mathsf{A})$ will be referred to as **simple definitions** for the abbreviation A and the synonym S respectively. An axiom of the form $\mathit{EquivalentClasses}(\mathsf{A}, \mathsf{C}_1, \ldots, \mathsf{C}_n)$ where $\mathsf{C}_1, \ldots \mathsf{C}_n$ are complex class expressions is a **ambiguous definition** of A. An axiom of the form $\mathit{EquivalentClasses}(\mathsf{S}_1, \ldots, \mathsf{S}_m)$ is an **enumerative definition** for the synonyms $\mathsf{S}_1, \ldots, \mathsf{S}_n$. And lastly, an axiom of the form $\mathit{EquivalentClasses}(\mathsf{S}_1, \ldots, \mathsf{S}_m, \mathsf{C}_1, \ldots, \mathsf{C}_n)$ will be referred to as a **compound definition** for $\mathsf{S}_1, \ldots, \mathsf{S}_m$, which are both synonyms and abbreviations.

With this notion of definition types, we can quantify how abbreviations and synonyms are specified explicitly in an ontology. However, counting implicit definitions of abbreviations and synonyms is not as straightforward, as extracting finite sets of entailments is a non-trivial matter [7]. We will delve into the determination and counting of implicit abbreviations and synonyms in more detail in Section 4. Before that, though, we address the more obvious question of how abbreviations and synonyms are *used* in an ontology.

Consider the example ontology $\mathcal{O}_{Ex}$ shown in Figure 1. Here, the abbreviation SpicyPizza is specified via a simple definition in axiom α_1 and occurs on the right-hand side of α_4. So, we say an abbreviation is **used** if it occurs in an OWL axiom that is not its definition. In addition to the *use* of an abbreviation, we can also determine if an abbreviation is *not used* even though its use would be possible. We refer to such a case as an abbreviation's **possible use**. For example,

[1]The Oxford English Dictionary defines the word *synonym* to denote "Strictly, a word having the same sense as another (in the same language); [...]" [6].

consider axiom $\alpha_1 \in \mathcal{O}_{Ex}$. Here, the abbreviation $\mathsf{SpicyTopping}$ (and its synonym $\mathsf{HotTopping}$) has *possible uses* since the complex OWL expression $\mathsf{PizzaTopping}$ and ($\mathsf{hasSpiciness}$ some Hot) could be replaced by either $\mathsf{SpicyTopping}$ or $\mathsf{HotTopping}$.

With the notions of an abbreviation's use and possible use, we can quantify the impact of abbreviations in an ontology. Before we do so, we come back to the topic of *determining* both explicit and implicit definitions of abbreviation and synonyms in an ontology.

4. Determining Abbreviations and Synonyms

Explicit definitions for abbreviations can be easily determined by checking the syntactic shape of all axioms in a given ontology. Similarly, implicit definitions can be determined by checking $\mathcal{O} \models \mathit{EquivalentClasses}(\mathsf{A}, \mathsf{C})$ for all pairs of named classes and complex classes occurring in an ontology. However, this becomes impractical for large ontologies with numerous named and complex classes.

Instead, to determine implicit abbreviations, we build upon highly optimized implementations of the standard reasoning service *classification*, i.e., computing all entailed $\mathit{SubClassOf}$ and $\mathit{EquivalentClasses}$ axioms between named classes in an ontology [8, 9, 10]. We will refer to this set as the inferred class hierarchy (ICH). The idea is to introduce an abbreviation for every complex class expressions that occurs in a given ontology, then to compute the ICH of the ontology with these newly added abbreviations, and finally to read off all implicit abbreviations from the ICH.

More formally, for a given ontology $\mathcal{O}$, we create the *abbreviation ontology*

$$\mathcal{O}_A = \mathcal{O} \cup \{\mathit{EquivalentClasses}(\mathsf{A}_i, \mathsf{C}_i) \mid \mathsf{C}_i \text{ occurs in } \mathcal{O}, \mathsf{A}_i \text{ does not occur in } \mathcal{O}\}$$

and compute $\mathsf{ICH}(\mathcal{O}_A)$. Since the ICH captures all $\mathit{SubClassOf}$ and $\mathit{EquivalentClasses}$ relationships between named classes in an ontology, it is straightforward to identify all named classes in $\mathcal{O}$ that are equivalent to a newly introduced abbreviation A_i in $\mathcal{O}_A$.

We will demonstrate this procedure by way of example. Consider the ontology $\mathcal{O}_{Ex}$ shown in Figure 1. This ontology contains complex class expressions $\mathsf{C}_1, \ldots, \mathsf{C}_6$ as shown in Figure 2a. Classifying the abbreviation ontology $\mathcal{O}^A_{Ex}$ and inspecting the ICH (see Figure 2) reveals that, for example, $\mathsf{SpicyTopping}$ is equivalent to A_2, which in turn is equivalent to C_2 by construction. So, $\mathsf{SpicyTopping}$ is an abbreviation for C_2 in $\mathcal{O}_{Ex}$.

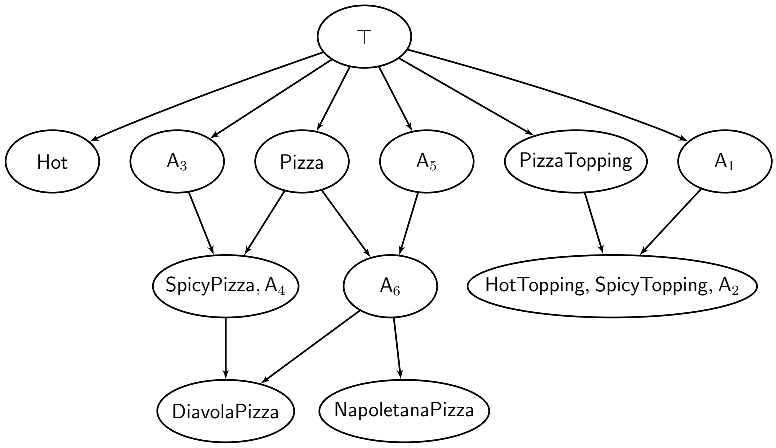
5. Study Design and Materials

Before we investigate to what extent abbreviations and synonyms are defined, used, and not used even though this would be possible, we first establish a baseline. This baseline aims to determine whether named classes in ontologies are associated with concrete domain-specific terms that are intended for human interpretation. Specifically, we assess the association of named classes in ontologies with human-readable annotations specified via `rdfs:label`[2] and `obo:definition`.[3] Similarly, we establish a baseline for human-readable synonyms specified

[2]https://www.w3.org/TR/rdf-schema/#ch_label
[3]http://purl.obolibrary.org/obo/IAO_0000115

C_1 = hasSpiciness some Hot
C_2 = PizzaTopping and (hasSpiciness some Hot)
C_3 = hasTopping some (PizzaTopping and (hasSpiciness some Hot))
C_4 = Pizza and (hasTopping some (PizzaTopping and (hasSpiciness some Hot)))
C_5 = hasCountryOfOrigin value Italy
C_6 = Pizza and hasCountryOfOrigin value Italy

(a) Complex class expressions in $\mathcal{O}_{Ex}$.



(b) Visualisation of ICH($\mathcal{O}^A_{Ex}$) without $\perp$.

Figure 2: Determining implicit abbreviations in $\mathcal{O}_{Ex}$ via the inferred class hierarchy of $\mathcal{O}^A_{Ex}$.

via `skos:altLabel`,[4] `oio:hasExactSynonym`, `oio:hasNarrowSynonym`, `oio:hasBroadSynonym`, `oio:hasRelatedSynonym`,[5] or `obo:alternativeLabel`.[6]

We conduct our empirical investigation using ontologies indexed in BioPortal as of February 2023.[7] The data set is created folllowing the same approach as described by Matentzoglu and Parsia [11] and includes a total of 785 ontologies. For orchestrating the empirical investigation, we use use the OWL API (v.5.1.15). We exclude ontologies that cannot be processed with the OWL API. Additionally, we exclude ontologies that do not contain any class expression axioms since such ontologies cannot contain abbreviations or synonyms. As a result of this procedure, our study corpus consists of 744 ontologies.

We group ontologies into three disjoint categories. First, ontologies that consist of atomic axioms only, i.e., *SubClassOf* and *EquivalentClasses* axioms that have only named classes as arguments. Second, ontologies expressible in $\mathcal{EL}^{++}$, and third, ontologies not expressible in $\mathcal{EL}^{++}$. We refer to these three kinds of ontologies as atomic, $\mathcal{EL}^{++}$, and rich ontologies

[4] https://www.w3.org/2012/09/odrl/semantic/draft/doco/skos_altLabel.html
[5] https://raw.githubusercontent.com/geneontology/go-ontology/master/contrib/oboInOwl#{hasExactSynonym, hasNarrowSynonym, hasBroadSynonym, hasRelatedSynonym}.
[6] http://purl.obolibrary.org/obo/IAO_0000118
[7] https://bioportal.bioontology.org/

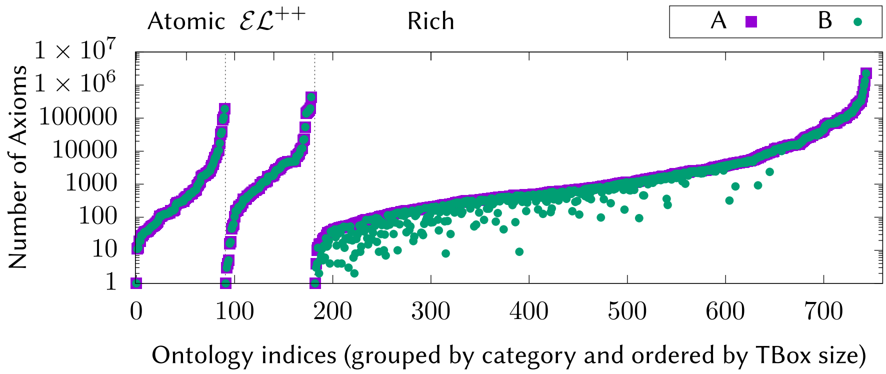


Figure 3: Size of (A) an ontology's TBox and (B) the set of an ontology's class expression axioms.

respectively. The study corpus contains 91 atomic ontologies, 88 $\mathcal{EL}^{++}$ ontologies, and 565 rich ontologies. We order ontologies within a category by the size of their TBoxes and assign each ontology an index in ascending order starting with atomic ontologies, then $\mathcal{EL}^{++}$ ontologies, and finally rich ontologies. Figure 3 illustrates this indexing by showing a comparison between the size of an ontology's (a) TBox and (b) the subset of class expression axioms.

Using the reasoner Konclude (v0.7.0-1138), we successfully classified 714 ontologies (for the purpose of determining implicit synonyms) and 656 abbreviation ontologies (for the purpose of determining implicit abbreviations).

6. Results

Before presenting our results on abbreviations and synonyms (see Section 3) we report on the use of annotation properties for specifying human-readable labels, definitions, and synonyms. Table 1 illustrates the number of ontologies that offer human-readable annotations for varying percentages of named classes.

We find that `rdfs:labels` are available in many ontologies for large proportions of named classes. For instance, $51 + 51 + 232 = 334$ ontologies provide `rdfs:labels` for all named classes. An additional $19 + 16 + 130 = 165$ ontologies provide `rdfs:labels` for at least 90% of named classes (but not 100%), so that $(334 + 165)/744 \approx 67\%$ of ontologies provide `rdfs:labels` for at least 90% of named classes. This provides strong evidence of the importance of human-readable `rdfs:labels` for named classes representing domain-specific concepts in biomedical ontologies.

We also find that `obo:definitions` are used in many ontologies. For example, 226 ontologies, i.e., $226/744 \approx 30\%$, provide `obo:definitions` for at least half of all named classes. While these proportions are smaller compared to `rdfs:labels`, they are non-trivial and provide evidence that `obo:definitions` play an important role in many biomedical ontologies.

However, human-readable annotations for synonyms appear to be less common in biomedical ontologies compared to `rdfs:labels` and `obo:definitions`. While there are a few ontologies

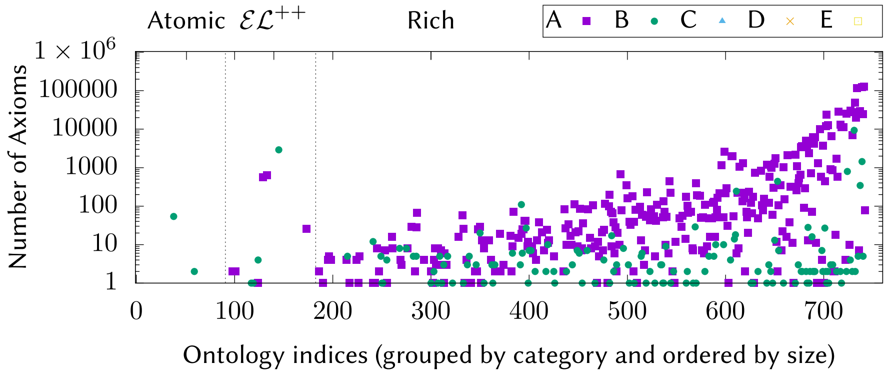


Figure 4: Number of (A) simple abbreviation definitions, (B) simple synonym definitions, (C) ambiguous abbreviation definitions, (D) enumerative synonym definitions, and (E) compound definitions.

Table 1

Number of atomic (A), $\mathcal{EL}^{++}$(E), and rich (R) ontologies that provide human-readable annotations for different proportions of named classes.

%	**Number of Ontologies**																							
	rdfs:label			**obo:def**			**obo:alt**			**skos:alt**			**oio:exact**			**oio:narrow**			**oio:broad**			**oio:related**		
	A	E	R	A	E	R	A	E	R	A	E	R	A	E	R	A	E	R	A	E	R	A	E	R
100	51	51	232	4	9	11	-	-	-	1	1	-	2	-	1	-	-	-	-	-	-	-	-	-
≥ 90	19	16	130	5	22	57	-	-	1	5	1	4	-	-	1	-	-	-	-	-	1	-	-	1
≥ 50	4	5	61	4	8	106	-	-	5	1	1	7	3	3	30	-	-	1	-	-	-	2	4	6
≥ 20	-	2	19	3	2	31	-	-	23	-	5	13	5	7	67	-	-	5	-	-	1	2	12	42
> 0	1	1	30	4	9	41	1	1	96	4	-	27	10	27	69	5	9	96	4	10	87	12	21	103
0	16	13	93	71	38	319	90	87	440	80	80	514	71	51	397	86	79	463	87	78	476	75	51	413

that annotate more than 90% of named classes with synonyms, e.g., 12 in the case of `skos:alt`, a lot of ontologies do not provide such synonym annotations for named classes at all (see last row in Table 1). This suggests that although annotations for synonyms are used in some biomedical ontologies, they do not seem to hold the same level of importance as `rdfs:labels` and `obo:definitions` for the most part.

The last observation can also be made w.r.t. the logical notions of abbreviations and synonyms. Figure 4 shows how many $\mathit{EquivalentClassesAxiom}$s are syntactic definition types for abbreviations or synonyms (see Section 3 for definition types). It becomes evident that there are about twice as many ontologies in which abbreviations are (explicitly) defined compared with ontologies in which synonyms are (explicitly) defined — namely 309 and 136 respectively. We also note that abbreviations and synonyms are specified only via simple definitions.

The difference between abbreviations and synonyms is not only evident in the *number of ontologies* in which they are defined but also in the *number of definitions* within ontologies. We find that the definitions for abbreviations are more numerous compared to definitions for synonyms. Specifically, there are 105 ontologies that define more than a hundred abbreviations, whereas only eight ontologies have more than a hundred definitions for synonyms. Given these observations, we will focus on abbreviations rather than synonyms in the remainder of this paper and will start with a discussion of explicitly defined abbreviations and then proceed with implicitly defined ones.

Figure 5 shows how many abbreviations are defined in ontologies. We observe that explicitly defined abbreviations (represented by green dots in Subfigure 5a) can be found in $309/744 \approx 41\%$ of ontologies. Furthermore, a considerable number of these ontologies contain numerous abbreviations, with 48 of them having at least a thousand explicitly defined abbreviations. Additionally, we notice that each explicitly defined abbreviation tends to be used (as shown by the blue triangles, indicating the number of used abbreviations, on top of the green dots, indicating the number of defined abbreviations in Subfigure 5a). Specifically, in 248 out of the 309 ontologies with explicitly defined abbreviations, all abbreviations are also used.

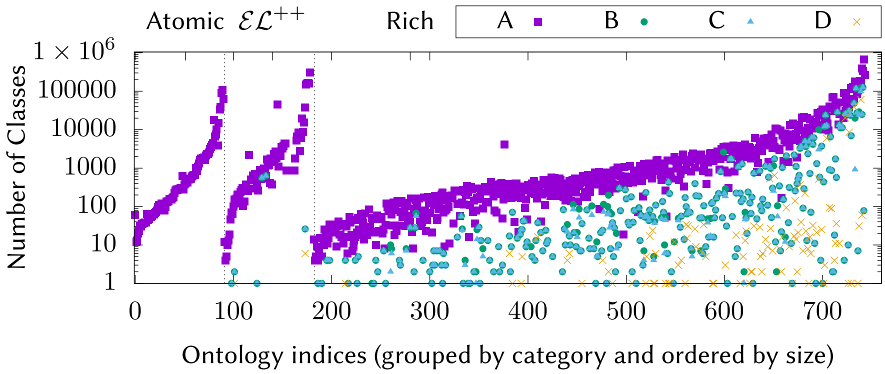
On the contrary, the number of explicitly defined abbreviations with potential uses tends to be considerably smaller compared to the overall number of defined abbreviations (indicated by the yellow cross in Subfigure 5a). Only one-third, specifically 101 out of 309 ontologies ($101/309 \approx 33\%$), contain explicitly defined abbreviations with possible uses. Additionally, it is important to note that no ontology exists where *all* explicitly defined abbreviations have potential uses. These observations suggest that explicitly defined abbreviations are generally used whenever possible, but there are a few exceptions in which over a thousand explicitly defined abbreviations have potential uses.

Regarding implicit abbreviations, there are 231 ontologies that define at least one abbreviation implicitly. It seems that ontologies generally have fewer implicitly defined abbreviations compared to explicitly defined ones. For instance, there are only 36 ontologies with more than a hundred implicitly defined abbreviations. However, it is important to note that implicit abbreviations could not be computed for 88 ontologies, and this group include many larger ontologies.

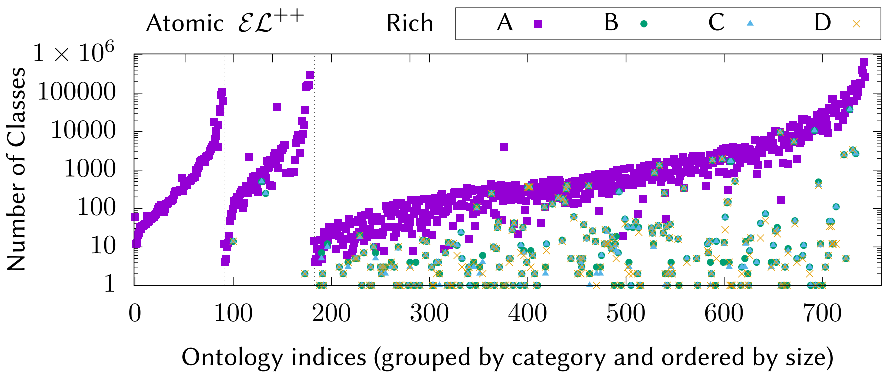
Before presenting the number of uses and possible uses for implicitly defined abbreviations, we remind the reader of the definition of an abbreviation's use in an ontology. An abbreviation's use is considered as its occurrence outside of its definition. Now, an implicitly defined abbreviation necessarily occurs in the ontology but there is no explicit definition. So, any implicitly defined abbreviations is also used (the only exceptions being `owl:Thing` and `owl:Nothing`). Regarding possible uses, we find that almost all implicitly defined abbreviations come with possible uses. In other words, most implicitly defined abbreviations are not used in at least one case where it would be possible to use them.

Besides counting *how many* abbreviations are defined, used, or not used, we are also interested in the question of *how often* abbreviations are used or could possibly be used. Since reporting these numbers for all abbreviations defined in all ontologies would be impractical (considering that some ontologies contain several thousand abbreviations), we focus on reporting the data for each ontology regarding the abbreviations with the *most uses* and *most possible uses.* Figure 6 depicts these numbers for both explicit and implicit abbreviations. We find that the abbreviation with the highest use for both explicitly and implicitly defined abbreviations (represented with a purple square and blue triangle respectively in Figure 6) tends to fall between 10 and 100 in most ontologies. However, in some large ontologies, this number can be much higher, reaching several thousand.

In the case of most possible uses, we find that the numbers for implicit abbreviations (represented with a yellow cross in Figure 6) are larger compared to the numbers of explicit ones (represented with a green dot). This provides additional evidence that explicit abbreviations tend to be used where possible, whereas implicitly abbreviations are not consistently reused.



(a) Explicitly defined abbreviations.



(b) Implicitly defined abbreviations.

Figure 5: Number of (A) named classes, (B) classes defined as abbreviations, (C) used abbreviations, (D) abbreviations with possible uses.

7. Related Work

Logical equivalent rewritings for ontologies are usually motivated for the purpose of improved reasoning performance [12] or ontology-based data access [13]. However, the idea of rewriting axioms to improve ontology comprehension has also been discussed. Existing work in this direction focuses on rewritings that are minimal in size because large expressions are arguably hard to read and comprehend [14, 15, 1]. Yet, it is debatable whether the smallest possible logical rewriting of an axiom is indeed most suitable for human interpretation.

In the work presented in this paper, the focus is not on rewritings that are minimal in size. Rather, we study to what extent domain-specific vocabulary defined in an ontology can be *reused* to simplify otherwise complex expressions. The main argument being that a meaningful

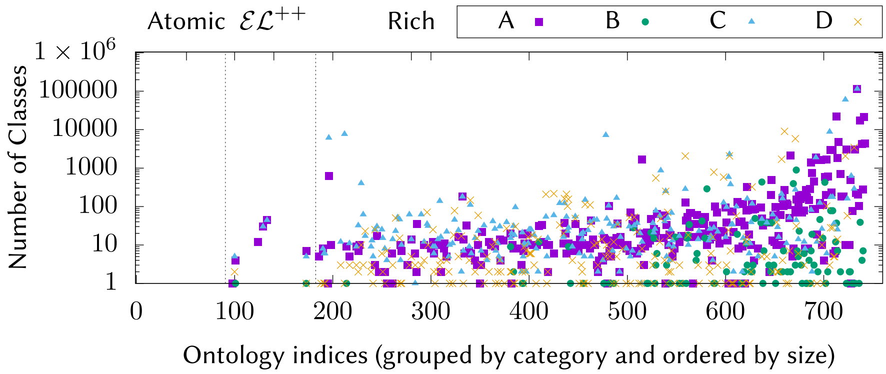


Figure 6: Number of maximal (A) explicit abbreviation use, (B) explicit abbreviation possible use, (C) implicit abbreviation use, (D) implicit abbreviation possible use.

name is more readily understood by domain experts compared to more complex expressions in OWL. It is important to note that the associated reduction in size is secondary in this context.

The task of determining abbreviations in an ontology (cf. Section 4) can be interpreted as concept definability, i.e., the problem of finding a definition for a concept name in an ontology [16]. However, we restrict the problem to finding definitions for concepts in terms of complex class expressions that already occur, syntactically speaking, in an ontology. Nevertheless, advances in research on concept definability may provide useful insights, e.g., knowing under what conditions implicitly defined concepts can also be defined explicitly.

8. Discussion & Future Work

The OBO Foundry considers naming conventions important for ontology comprehension, readability, navigability, alignment, and integration [17] and recommends that the majority of classes in an ontology should have textual definitions [18]. While not all biomedical ontologies conform to the principles put forward by the OBO Foundry, there is no question that human-readable names and definitions are used in many ontologies (see Table 1 in Section 6).

The use of human-readable names is important, because technical terms in a domain-specific vocabulary tend to be defined in terms of already defined terms. For example, a ’blood assay datum’ is defined as ’A data item that is the specified output of a blood assay’. This definition only makes sense if the notion of a ’blood assay’ is already defined. So, if textual definitions make use of already defined terms, and textual definitions should match logical definitions, as the OBO Foundry advocates, then possible uses of explicitly defined abbreviations (cf. Section 3) should not occur.[8]

However, our results suggest that even though the reuse of already defined concepts seems to be preferred, there is a non-trivial number of cases in which a complex class expression

[8]See https://obofoundry.org/principles/fp-006-textual-definitions.html.

could be replaced by an existing equivalent named class. It would be interesting to consult with the ontology developers in such cases to determine whether such cases are intended or not. Likewise, it would be interesting to find out whether classes with *implicit* logical definitions are intentional and should be made explicit, and whether they should be reused whenever possible.

In addition to the question of *when to use* an abbreviation, is the question of *when to introduce* a new abbreviation. In particular, if a complex class expression occurs often in an ontology, one may want to think about whether such an expression can be given a meaningful name and which should be used instead.

However, it needs to be highlighted that the introduction of an abbreviation, as defined in this work, *changes the meaning* of an ontology. Consider the ontology $\mathcal{O}$ and $\mathcal{O}_A = \mathcal{O} \cup \{\alpha\}$ where $\alpha = \mathit{EquivalentClasses}(\mathsf{A}, \mathsf{C})$ is a definition for an abbreviation A. If A does not occur in $\mathcal{O}$, then $\mathcal{O} \not\equiv \mathcal{O}_A$ because $\mathcal{O}_A \models \alpha$ but $\mathcal{O} \not\models \alpha$. This change in meaning can be avoided by encoding abbreviations using a meta-language, e.g., OTTR [19], on top of OWL. As an example, consider the ontology

$$\mathcal{O} = \{ \begin{array}{ll} \mathsf{Napoletana} & \mathsf{SubClassOf\ Pizza\ and\ hasCountryOfOrigin\ value\ Italy}, \\ \mathsf{Diavola} & \mathsf{SubClassOf\ Pizza\ and\ hasCountryOfOrigin\ value\ Italy}, \\ \mathsf{Hawaiian} & \mathsf{SubClassOf\ Pizza\ and\ hasCountryOfOrigin\ value\ Canada} \end{array} \}.$$

With OTTR, a mapping `ItalianPizza` $\mapsto$ Pizza and hasCountryOfOrigin value Italy can be defined, so that $\mathcal{O}$ can be encoded as

$$\mathcal{O}_T = \{ \begin{array}{ll} \mathsf{Napoletana} & \mathsf{SubClassOf}\ \mathtt{ItalianPizza}, \\ \mathsf{Diavola} & \mathsf{SubClassOf}\ \mathtt{ItalianPizza}, \\ \mathsf{Hawaiian} & \mathsf{SubClassOf\ Pizza\ and\ hasCountryOfOrigin\ value\ Canada} \end{array} \}.$$

Note that `ItalianPizza` is not an OWL class but an expression in OTTR. In particular, the ontology $\mathcal{O}$ is semantically equivalent to $\mathcal{O}_T$ because the OTTR expression `ItalianPizza` is *indistinguishable* from Pizza and hasCountryOfOrigin value Italy on the level of OWL. The use of a meta-level language also opens up possibilities to capture definitions on higher level of abstraction than OWL. In the case of the example ontology $\mathcal{O}$, the representation of a pizza's country of origin could be captured by a *parameterized* OTTR expression `PizzaWithOrigin`$(x) \mapsto$ Pizza and hasCountryOfOrigin value x. With this, all three pizzas in $\mathcal{O}$ can be encoded in a uniform manner giving rise to the following even more meaningful definitions:

$$\mathcal{O}_P = \{ \begin{array}{ll} \mathsf{Napoletana} & \mathsf{SubClassOf}\ \mathtt{PizzaWithOrigin}(\mathsf{Italy}), \\ \mathsf{Diavola} & \mathsf{SubClassOf}\ \mathtt{PizzaWithOrigin}(\mathsf{Italy}), \\ \mathsf{Hawaiian} & \mathsf{SubClassOf}\ \mathtt{PizzaWithOrigin}(\mathsf{Canada}) \end{array} \}.$$

9. Conclusion

In this paper, we proposed an approach for analyzing and quantifying the use of logical abbreviations, i.e., named classes that are defined to be logically equivalent to complex class expressions. We used this approach to survey biomedical ontologies indexed in BioPortal and find that abbreviations are highly prevalent. Although there are some exceptions, *explicitly* defined abbreviations tend to be used whenever possible. However, *implicitly* defined abbreviations often come with many possible uses which rasies the question of whether this is intentional or undesireable.

References

[1] N. Nikitina, P. Koopmann, Small Is Beautiful: Computing Minimal Equivalent EL Concepts, in: AAAI, AAAI Press, 2017, pp. 1206–1212.

[2] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, U. Sattler, OWL 2: The next step for OWL, Journal of Web Semantics 6 (2008) 309–322.

[3] B. Motik, P. Patel-Schneider, B. Parsia, OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition) (2012). URL: http://www.w3.org/TR/owl2-syntax/.

[4] M. Horridge, P. Patel-Schneider, OWL 2 Web Ontology Language Manchester Syntax (Second Edition) (2012). URL: https://www.w3.org/TR/owl2-manchester-syntax/.

[5] abbreviation, n., in: OED Online, Oxford University Press, 2022. URL: https://www.oed.com/view/Entry/180?redirectedFrom=abbreviation&, accessed July 05, 2022.

[6] synonym, n., in: OED Online, Oxford University Press, 2022. URL: https://www.oed.com/view/Entry/196522?result=1&rskey=EVWsim&, accessed July 05, 2022.

[7] S. Bail, B. Parsia, U. Sattler, Extracting Finite Sets of Entailments from OWL Ontologies, in: Description Logics, volume 745 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011.

[8] F. Baader, B. Hollunder, B. Nebel, H. Profitlich, E. Franconi, An Empirical Analysis of Optimization Techniques for Terminological Representation Systems, or Making KRIS Get a Move on, in: KR, Morgan Kaufmann, 1992, pp. 270–281.

[9] B. Glimm, I. Horrocks, B. Motik, R. D. C. Shearer, G. Stoilos, A novel approach to ontology classification, J. Web Semant. 14 (2012) 84–101.

[10] Y. Kazakov, M. Krötzsch, F. Simancik, The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with $\mathcal{EL}$ Ontologies, J. Autom. Reason. 53 (2014) 1–61.

[11] N. Matentzoglu, B. Parsia, BioPortal Snapshot 30.03.2017, 2017. URL: https://doi.org/10.5281/zenodo.439510. doi:10.5281/zenodo.439510.

[12] D. Carral, C. Feier, B. C. Grau, P. Hitzler, I. Horrocks, *EL*-ifying Ontologies, in: IJCAR, volume 8562 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 464–479.

[13] M. Imprialou, G. Stoilos, B. C. Grau, Benchmarking Ontology-Based Query Rewriting Systems, in: AAAI, AAAI Press, 2012.

[14] F. Baader, R. Küsters, R. Molitor, Rewriting Concepts Using Terminologies, in: KR, Morgan Kaufmann, 2000, pp. 297–308.

[15] M. Horridge, B. Parsia, U. Sattler, Laconic and Precise Justifications in OWL, in: ISWC, volume 5318 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 323–338.

[16] B. ten Cate, E. Franconi, I. Seylan, Beth Definability in Expressive Description Logics, J. Artif. Intell. Res. 48 (2013) 347–414.

[17] D. Schober, B. Smith, S. E. Lewis, W. Kusnierczyk, J. Lomax, C. Mungall, C. F. Taylor, P. Rocca-Serra, S. Sansone, Survey-based naming conventions for use in OBO Foundry ontology development, BMC Bioinform. 10 (2009). URL: https://doi.org/10.1186/1471-2105-10-125. doi:10.1186/1471-2105-10-125.

[18] R. C. Jackson, N. Matentzoglu, J. A. Overton, R. Vita, J. P. Balhoff, P. L. Buttigieg, S. Carbon, M. Courtot, A. D. Diehl, D. M. Dooley, W. D. Duncan, N. L. Harris, M. A. Haendel, S. E. Lewis, D. A. Natale, D. Osumi-Sutherland, A. Ruttenberg, L. M. Schriml, B. Smith, C. J. S. Jr., N. A. Vasilevsky, R. L. Walls, J. Zheng, C. J. Mungall, B. Peters, OBO Foundry in 2021: operationalizing open

data principles to evaluate ontologies, Database J. Biol. Databases Curation 2021 (2021). URL: https://doi.org/10.1093/database/baab069. doi:10.1093/database/baab069.
[19] M. G. Skjæveland, D. P. Lupp, L. H. Karlsen, H. Forssell, Practical Ontology Pattern Instantiation, Discovery, and Maintenance with Reasonable Ontology Templates, in: ISWC (1), volume 11136 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 477–494.