24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Applying the Closed World Assumption to SUMO-based FOL Ontologies for Effective Commonsense Reasoning

Javier Álvez[1] and **Itziar Gonzalez-Dios**[2] and **German Rigau**[3]

Abstract. Most commonly, the *Open World Assumption* is adopted as a standard strategy for the design, construction and use of ontologies. This strategy limits the inferencing capabilities of any system because non-asserted statements (*missing knowledge*) could be assumed to be alternatively true or false. As we will demonstrate, this is especially the case of first-order logic (FOL) ontologies where non-asserted statements is nowadays one of the main obstacles to its practical application in automated commonsense reasoning tasks. In this paper, we investigate the application of the *Closed World Assumption* (CWA) to enable a better exploitation of FOL ontologies by using state-of-the-art automated theorem provers. To that end, we explore different CWA formulations for the structural knowledge encoded in a FOL translation of the SUMO ontology, discovering that almost 30 % of the structural knowledge is missing. We evaluate these formulations on a practical experimentation using a very large commonsense benchmark obtained from WordNet through its mapping to SUMO. The results show that the competency of the ontology improves more than 50 % when reasoning under the CWA. Thus, applying the CWA automatically to FOL ontologies reduces their ambiguity and more commonsense questions can be answered.

1 Introduction

Large knowledge-bases and complex ontologies are being used in a wide range of knowledge based systems [7] that require practical commonsense reasoning [8, 28, 39, 10, 43]. To represent this knowledge, the most prominent and fundamental logical formalism is the *first-order predicate calculus*, or first-order logic (FOL) for short. The semantics of FOL, and thus also of *Description Logics* (DL), operates under the *Open World Assumption* (OWA) allowing monotonic reasoning [11]. OWA considers that statements which are *not* logical consequences of a given knowledge base are not necessarily considered false but possible. Therefore, statements that are false or impossible must be clearly stated as so in the ontology. The OWA presumes incomplete knowledge about the domain being modelled. Thus, ontologies basically encode positive information about the modelled world since the number of negative facts vastly exceeds the number of positive ones. In fact, under the OWA, it is totally unfeasible to explicitly represent all such negative information in the ontology.

Otherwise, the *Closed World Assumption* (CWA) presumes perfect knowledge about the domain being modelled. CWA is a common non-monotonic technique that allows to deal with negative information in knowledge bases and data bases [34]. In fact, commonsense reasoning is non-monotonic: the addition of new knowledge can invalidate conclusions drawn before the addition [17].

There is a considerable computational and representational advantage to reason under the CWA since negative information should be inferred by default [35]. The Careful CWA (CCWA) is an extension of the CWA [15]. It allows us to restrict the effects of closing the world by specifying the predicates which may be affected by the CWA rule in indefinite databases.

Nowadays OWL 2 [44] is currently one of the most common formal knowledge representation formalism, but it is unable to fully cope with general upper ontologies like Cyc [24], DOLCE [14] or SUMO [26] since full FOL expressivity or higher is required. Further, CWA cannot be entirely applied to DL ontologies, but some approximations have been proposed in the literature [12, 25, 22].

In order to provide advanced reasoning support to large FOL conversions of expressive ontologies [33, 20, 30, 4] state-of-the-art automated theorem provers (ATP) for FOL such as Vampire [21] or E [36] have proven its efficiency by implementing many sophisticated techniques like axiom selection [19]. However, the semi-decidability of FOL and the poor scalability of the known decision procedures have been usually identified as the main drawbacks for the practical use of FOL ontologies.

In this paper, we report on our empirical research applying the Careful CWA, that was originally conceived for indefinite databases, to the structural knowledge of a FOL ontology. In particular, we propose two complementary strategies for the application of the CCWA to the structural knowledge about classes represented in a FOL conversion of the top levels of SUMO [26]. To the best of our knowledge, this is the first attempt of applying the CWA to FOL ontologies since up to now the research on SUMO has been developed under the OWA.

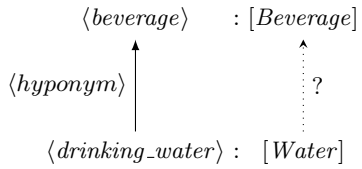

Figure 1. An example of competency question for SUMO obtained from WordNet

We test the original and the resulting versions of SUMO by using the knowledge in WordNet [13] as gold standard. For this pur-

[1] LoRea Group, University of the Basque Country (UPV/EHU), email: javier.alvez@ehu.eus
[2] Ixa Group, HiTZ Center, University of the Basque Country (UPV/EHU), email: itziar.gonzalezd@ehu.eus
[3] Ixa Group, HiTZ Center, University of the Basque Country (UPV/EHU), email: german.rigau@ehu.eus

pose, we build a benchmark by automatically deriving a very large set of *competency questions* (CQs) from WordNet and its mapping to SUMO [27] on the basis of 7 manually created *question patterns (QPs)* [5]. These QPs focus on the main structural relations of WordNet, which are *hyponymy* and *antonymy*. The results show that applying carefully the CWA to the structural knowledge about classes in SUMO improves the competency of the ontology more than 50 % when reasoning on the same commonsense benchmark.

For instance, in Figure 1 we describe the CQ "*Drinking water is a beverage*" that results from the hyponymy pair of WordNet synsets *drinking_water* (the *hyponym*) and *beverage* (the *hyperonym*), which are respectively connected to the SUMO concepts *Water* and *Beverage*. From this CQ, we obtain two conjectures: the first one states that some instances of *Water* can be instance of *Beverage*, and the second one is its negation. None of these conjectures are solved using SUMO, but one of them is entailed depending on the strategy for the application of the CWA to SUMO, concretely the CCWA to *subclass* and *disjoint* predicates. From now on, we will refer to CCWA as CWA.

Our research empirically demonstrates the existence of large knowledge gaps in SUMO and that the *missing knowledge* in SUMO-based FOL ontologies is nowadays one of the main obstacles for its practical application in automated commonsense reasoning tasks. Anyway, our proposal is not intended to provide that missing knowledge but, nevertheless, it could help ontologists to complete the knowledge in the ontology.

The contributions of this paper are fourfold. First, we propose an effective method to apply automatically the CWA to the structural knowledge of SUMO, which enables a really compact formalization. Second, we perform a detailed analysis of the empirical results obtained when comparing the resulting versions of SUMO with the original one on a very large commonsense benchmark with more than 14,000 CQs. This analysis demonstrates that the competency of the ontology can improve more than 50 % when reasoning under the CWA. Third, we provide a quantitative analysis of the structural knowledge about classes that still needs to be encoded in the top levels of SUMO. Fourth, we discuss some suitable design criteria that enable the automated application of the CWA in FOL ontologies.

Outline of the paper. In Section 2 we present SUMO and its translations into FOL; in Section 3, we describe our approaches for the application of the CWA to *subclass* and *disjoint*; in Section 4 we report on the experimental results that we discuss in Section 5; and we conclude in Section 6 by outlining the future work.

2 SUMO and its FOL versions

SUMO[4] [26] is a well-known upper level ontology proposed as a starter document by the IEEE Standard Upper Ontology Working Group. SUMO is expressed in SUO-KIF (Standard Upper Ontology Knowledge Interchange Format [29]), which is a dialect of KIF (Knowledge Interchange Format [16]). The syntax of both KIF and SUO-KIF goes beyond FOL and, therefore, SUMO axioms cannot be directly used by FOL ATPs without a suitable transformation.

To the best of our knowledge, there are two main proposals for the translation of the two upper levels of SUMO into FOL formulas that are described in [30, 31] and [4] respectively. Both proposals have been developed under the OWA and are currently included in the *Thousands of Problems for Theorem Provers* (TPTP) problem library[5] [40].

The knowledge of SUMO, and therefore of its translations into FOL, is organized around the notions of *classes* and *particulars*. The main structural knowledge about classes is provided by the predicates i) *subclass*, which is defined as a partial order relation (reflexive, transitive and anti-symmetric), and ii) *disjoint*, which is defined as symmetric and irreflexive. The predicate *subclass* provides the classical concept of relation inclusion between classes, while the predicate *disjoint* relates incompatible classes: in Adimen-SUMO, incompatible classes cannot share any common instance or subclass. In SUMO, particulars are introduced by the predicate *instance*.

From the axiomatization of SUMO, particulars are inherited by superclasses (inheritance of *instance* via *subclass*). Additionally, every pair of disjoint classes do not share any instance and are not subclass of each other. Further, SUMO includes some additional predicates that provide structural knowledge about disjoint classes: specifically, *partition* and *disjointDecomposition*.

In the experiments, we will use Adimen-SUMO the FOL version of SUMO that has proved to be more competent [1, 6, 37]. Currently, Adimen-SUMO consists of 8,291 formulas, out of them 5,255 are atomic, and defines $2,169$ classes.

Since Adimen-SUMO has been developed under the OWA, sometimes negative knowledge is not inferable. For example, although considering both the explicit and implicit knowledge in Adimen-SUMO, it is not possible to infer whether *SentientAgent* ("*An agent that has rights but may or may not have responsibilities and the ability to reason*") and *Sandwich* ("*Any food which consists of two or more pieces of bread and some sort of filling between the two pieces of bread*") are related by *subclass*/*disjoint* or not. From now on, we say that pairs of classes are *non-asserted pairs* (or *missing knowledge*) if Adimen-SUMO cannot entail whether they are related or not by *subclass*/*disjoint*. We have obtained an upper-bound of the amount of missing structural knowledge in Adimen-SUMO. For this purpose, we have considered both the explicit and implicit knowledge in Adimen-SUMO as follows: first, we have used *ad hoc* tools by focusing on the structural knowledge from Adimen-SUMO; second, we have used ATPs with the whole knowledge of Adimen-SUMO. Among the total of $4,704,561$ ($2,169^2$) different pairs of classes, it is possible to infer that a) $18,374$ (0.39 %) pairs of classes are related by *subclass* (thus, not related by *disjoint*), b) $3,304,246$ (70.23 %) pairs of classes are related by *disjoint* (thus, not related by *subclass*) and c) $62,069$ (1.32 %) pairs of classes are not related by *disjoint* (because those pairs of classes share some instance/subclass). Consequently, there are at most $1,381,941$[6] non-asserted *subclass* pairs (29.37 %) and $1,338,246$[7] non-asserted *disjoint* pairs (28.45 %). In other words, in the worst case scenario almost 30 % of the structural knowledge about classes is missing in Adimen-SUMO.

3 Completing Adimen-SUMO

In this section, we describe different applications of the CWA to the structural knowledge about classes in Adimen-SUMO in order to reduce missing knowledge. We also provide the amount of new formulas that are required in each case. Concretely, we focus on the predicates *subclass* and *disjoint*. In the case of *subclass*, we apply a single strategy in which every non-asserted *subclass* pairs in Adimen-SUMO are assumed not to be related by *subclass* (see Subsection 3.1). With respect to *disjoint*, we apply two complementary strategies by assuming disjointness/non-disjointness to non-asserted pairs (see Subsections 3.2 and 3.3 respectively).

[4] `http://www.ontologyportal.org`
[5] `http://www.tptp.org`

[6] $1,381,941 = 4,704,561 - (18,374 + 3,304,246)$
[7] $1,338,246 = 4,704,561 - (3,304,246 + 62,069)$

3.1 Applying the CWA to *subclass*

In this subsection, we describe our proposal for the application of the CWA to *subclass* by assuming that the non-asserted *subclass* pairs are not related by *subclass*.

The application of the CWA to *subclass* is based on the set of class pairs that are explicitly related by *subclass*: *direct* subclasses. From now on, we denote by $all\_direct\_subclasses\_of(c)$ the set of all the Adimen-SUMO classes that are explicitly defined to be direct subclasses of an Adimen-SUMO class c.

In order to apply the CWA to *subclass*, we conveniently adapt the *data base completion* method proposed in [9]. For this adaptation, we implicitly adopt the *Domain Closure Asssumption* (DCA) (that is, closed domain of classes) and assume that the domain of classes is finite: the domain only includes the classes that are explicitly introduced by the ontology.

Our adaptation of the data base completion relies on the fact that *subclass* is defined as a partial order relation in SUMO. This implies that given two classes c and c' such that c' is direct subclass of c (that is, $c' \in all\_direct\_subclasses\_of(c)$), every subclass of c' is also subclass of c (by transitivity) and c is subclass of itself (by reflexivity):

$$\forall ?x \forall ?y \forall ?z \ (\ subclass(?x, ?y) \wedge subclass(?y, ?z)) \rightarrow \quad (1)$$
$$subclass(?x, ?z)\ )$$
$$\forall ?x \ (\ subclass(?x, ?x)\ ) \quad (2)$$

Further, we also know that any superclass of c (except of c itself) is not subclass of c (by antisymmetry):

$$\forall ?x \forall ?y \ (\ (subclass(?x, ?y) \wedge subclass(?y, ?x)) \rightarrow x = y\ ) \quad (3)$$

Consequently, the *completed set of subclasses of an Adimen-SUMO class c* can be defined as

$$\forall ?x \ (\ subclass(?x, c) \leftrightarrow (?x = c \vee \bigvee_{i=1}^{n} subclass(?x, c_i))\ ) \quad (4)$$

where $all\_direct\_subclasses\_of(c) = \{c_1, \ldots, c_n\}$. The reverse implication is already given by the ontology. In particular, by axioms (1-3). Thus, we only have to augment Adimen-SUMO by including the direct implication. For example, *BloodCell* have two direct subclasses in Adimen-SUMO, which are *RedBloodCell* and *WhiteBloodCell*:

$$subclass(RedBloodCell, BloodCell) \quad (5)$$
$$subclass(WhiteBloodCell, BloodCell) \quad (6)$$

Therefore, we have that $all\_direct\_subclasses\_of(BloodCell) = \{RedBloodCell, WhiteBloodCell\}$ and the formula that results from (4) to complete the information about the subclasses of *BloodCell* in Adimen-SUMO is:

$$\forall ?x \ (\ subclass(?x, BloodCell) \rightarrow \quad (7)$$
$$(\ ?x = BloodCell \ \vee$$
$$subclass(?x, RedBloodCell) \ \vee$$
$$subclass(?x, WhiteBloodCell)\ )\ )$$

It is worth noting that, although the transitive closure of a binary relation cannot —in general— be expressed in first-order logic, all the formulas involved in the proposed application of the DCA to Adimen-SUMO classes (axioms (1-3) and (5-7) in the case of *BloodCell*) are pure FOL formulas.

In total, since Adimen-SUMO inherits $2,169$ classes from SUMO we have automatically augmented Adimen-SUMO by including $2,169$ new formulas as the one above (one per SUMO class), where we have used $4,705$ subclass atoms.

In this approach every non-asserted *subclass* pairs in Adimen-SUMO are assumed not to be related by *subclass*. It is worth noting that the complementary strategy —that is, assuming that every non-asserted *subclass* pairs in Adimen-SUMO are related by *subclass*— turns most of the classes into equal by antisymmetry.

3.2 Applying the CWA to *disjoint* by assuming disjointness

In this subsection, we describe our proposal for the application of the CWA to *disjoint* by assuming that the non-asserted *disjoint* pairs of classes are disjoint.

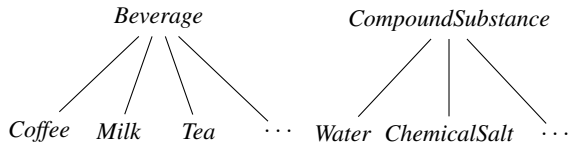

Figure 2. Non-asserted *disjoint* subclasses of *Beverage* and *CompoundSubstance*

Formally, the application of the CWA by assuming disjointness can be described as follows: for any pair of non-asserted *disjoint* classes c_1 and c_2, we augment Adimen-SUMO by stating that c_1 and c_2 are disjoint. For example, in Figure 2 we show some of the subclasses of *Beverage* and *CompoundSubstance*, where all the depicted subclasses of *Beverage* are non-asserted *disjoint* with all the depicted subclasses of *CompoundSubstance*. Hence, the above described application of the CWA to *disjoint* by assuming disjointness introduces, among others, the following formulas in Adimen-SUMO:

$$disjoint(Beverage, CompoundSubstance) \quad (8)$$
$$disjoint(Beverage, Water) \quad (9)$$

It is obvious that the second conjecture obtained from "*Drinking water is a beverage*" is entailed by the augmented version of Adimen-SUMO, because *Beverage* and *Water* do not have any common instance/subclass if they are disjoint (see formula (9)).

In practice, most of the formulas that results from the application of the above method are redundant because of the axiomatization of *disjoint* in SUMO and can be easily omitted. More specifically, two classes are related by *disjoint* iff there is no common instance or subclass. Hence, given a pair of disjoint classes c_1 and c_2, every subclass c_1' of c_1 (resp. c_2' of c_2) is disjoint with c_2 (resp. c_1) by the inheritance of *instance* via *subclass* and the transitivity of *subclass* since all the instances/subclasses of c_1' (resp. c_2') are also instance/subclass of c_1 (resp. c_2). Therefore, disjointness is inherited by subclasses. For example, the classes *Beverage* and *Water* are still inferred to be disjoint although augmenting Adimen-SUMO by only formula (8) (and not formula (9)), because *Water* is defined as subclass of *CompoundSubstance* in Adimen-SUMO.

This way, we have augmented Adimen-SUMO by adding $20,896$ non-redundant atomic formulas.

Table 1. Summary of Experimentation Results

Competency Questions	OWA			CWA-D			CWA-n-D		
	#	%	T	#	%	T	#	%	T
noun #1 (7,402)	4,016	54.26 %	47.86 s.	5,250	70.93 %	54.71 s.	5,064	68.41 %	51.28 s.
noun #2 (1,866)	1,209	64.79 %	12.87 s.	1,475	79.05 %	31.20 s.	1,288	69.02 %	19.09 s.
verb #1 (1,740)	637	36.61 %	72.79 s.	1,454	83.56 %	41.94 s.	1,340	77.01 %	51.42 s.
verb #2 (299)	144	48.16 %	30.92 s.	252	84.28 %	61.44 s.	155	51.84 %	23.42 s.
antonym #1 (65)	36	55.38 %	44.90 s.	29	44.62 %	33.99 s.	27	41.54 %	39.49 s.
antonym #2 (504)	152	30.16 %	108.15 s.	159	31.55 %	80.53 s.	111	22.02 %	110.85 s.
antonym #3 (2,448)	1,091	44.57 %	155.79 s.	1,162	47.47 %	99.50 s.	1,513	61.81 %	124.16 s.
Total (14,324)	**7,285**	**50.86 %**	**61.30 s.**	**9,781**	**68.28 %**	**55.12 s.**	**9,498**	**66.31 %**	**58.75 s.**

3.3 Applying the CWA to *disjoint* by assuming non-disjointness

Conversely, next we describe the application of the CWA to *disjoint* by assuming that the non-asserted *disjoint* pairs of classes are non-disjoint.

For this purpose, we proceed similar to our previous strategy: for any pair of non-asserted *disjoint* classes c_1 and c_2, augment Adimen-SUMO by stating that c_1 and c_2 are non-disjoint. Coming back to the example about the non-asserted *disjoint* classes in Figure 2, the above application of the CWA to *disjoint* by assuming non-disjointness would introduce, among others, the next formulas:

$$\neg disjoint(Beverage, CompoundSubstance) \quad (10)$$
$$\neg disjoint(Beverage, Water) \quad (11)$$
$$\neg disjoint(Coffee, Water) \quad (12)$$

This time, the first conjecture that results from "*Drinking water is a beverage*" is entailed by the augmented version of Adimen-SUMO due to formula (11), since pairs of non-disjoint subclasses necessarily have some common instance/subclass. However, this way we would obtain, as before, many redundant formulas: given a pair of non-disjoint classes c_1 and c_2, it can be inferred in Adimen-SUMO that all the superclasses c_1' of c_1 (resp. c_2' of c_2) are non-disjoint with c_2 (resp. c_1) by the inheritance of *instance* via *subclass* and the transitivity of *subclass*. That is, non-disjointness is inherited *upwards*. However, it is not easy to omit the redundant formulas in this case: for this purpose, we have to introduce non-disjoint pairs only between classes that do not have subclasses. This generates a very high number of new formulas. In addition, we need to check whether the classes are defined as disjoint. To verify this, two strategies can be followed: 1) stating for each pair if it is disjoint, for which it is necessary to use a lot of memory; 2) checking the hierarchy of classes. In this last case, there are two other options:

- By following a top-down strategy: given a branch, it is necessary to check all other non-disjoint ones, with many repetitions.
- By following a bottom-up strategy: for each pair that is non-disjoint, it is necessary to check the whole branch from the leaves to the root, which is computationally expensive.

In order to minimize the number of atomic formulas that are required to apply the CWA to *disjoint* by assuming non-disjointness, we introduce a new predicate —*nonDisjoint*— that states the *downwards* inheritance of non-disjointness. This new predicate is axiomatized as follows:

$$\begin{aligned}\forall ?x_1 \forall ?x_2 \forall ?y_1 \forall ?y_2 \; (\; (\; & nonDisjoint(?x_1, ?x_2) \wedge \\ & subclass(?y_1, ?x_1) \wedge \\ & subclass(?y_2, ?x_2) \;) \; \rightarrow \\ & \neg disjoint(?y_1, ?y_2) \;)\end{aligned} \quad (13)$$

By means of this new predicate, we can proceed as follows: for any pair of classes c_1 and c_2 such that

a) the pair c_1 and c_2 is non-asserted *disjoint*,
b) any subclasses c_1' and c_2' of c_1 and c_2 respectively are either non-asserted *disjoint* or not related by *disjoint*

then we augment Adimen-SUMO by stating that c_1 and c_2 are related by *nonDisjoint*. By proceeding this way, non-disjointness is inherited both upwards and downwards, which enables a very compact formalization. In the above example, *Beverage* and *Water* are non-asserted *disjoint* (condition *a*) and, additionally, all the subclasses of *Beverage* (*Coffee*, *Milk* and *Tea*, among others) are non-asserted *disjoint* or not related with all the subclasses of *Water* (condition *b*). Consequently, we augment Adimen-SUMO as follows by the application of the CWA to *disjoint* assuming non-disjointness:

$$nonDisjoint(Beverage, Water) \quad (14)$$

By formula (14), *Beverage* and *Water* are directly asserted to be non-disjoint, as given by formula (11). Further, all the pairs obtained from the super-classes of *Beverage* and *Water* respectively are asserted to be non-disjoint by upwards inheritance: for example, *Beverage* and *CompoundSubstance* (as given by formula (10)). Additionally, all the pairs obtained from the subclasses of *Beverage* and *Water* respectively are asserted to be non-disjoint by downwards inheritance: for example, *Beverage* and *Coffee* (as given by formula (12)), *Beverage* and *Milk*, etc. Consequently, formulas (10-12) (and many others) can be replaced with formulas (13-14) while preserving logical equivalence. It is worth noting that condition *b* prevents the introduction of inconsistencies caused by the downwards inheritance of *non-disjoint*, since all the involved pairs of subclasses are restricted to be non-asserted *disjoint* or non related by *disjoint*.

In total, we have augmented Adimen-SUMO by adding $29,643$ atomic formulas and 1 general formula.

4 Experimental Results

In this section, we present the experiments with the different versions of Adimen-SUMO under the OWA and CWA as introduced in Section 3.

Table 2. New solved CQs under the CWA

Competency Questions	Passing						Non-passing					
	CWA-D			CWA-n-D			CWA-D			CWA-n-D		
	#	%	T	#	%	T	#	%	T	#	%	T
noun #1 (3,386)	36	1.06 %	115.16 s.	1,311	38.72 %	64.89 s.	1,486	43.89 %	8.79 s.	4	0.12 %	69.78 s.
noun #2 (657)	3	0.46 %	3.70 s.	1	0.15 %	240.12 s.	288	43.84 %	101.60 s.	103	15.68 %	83.97 s.
verb #1 (1,103)	12	1.09 %	65.00 s.	735	66.64 %	61.27 s.	835	75.70 %	4.24 s.	0	0.00 %	0.00 s.
verb #2 (155)	0	0.00 %	0.00 s.	0	0.00 %	0.00 s.	113	72.90 %	120.67 s.	25	16.13 %	111.39 s.
antonym #1 (29)	1	3.45 %	3.74 s.	0	0.00 %	0.00 s.	1	3.45 %	10.25 s.	1	3.45 %	40.39 s.
antonym #2 (352)	34	9.66 %	67.61 s.	11	3.13 %	93.74 s.	1	0.28 %	3.72 s.	0	0.00 %	0.00 s.
antonym #3 (1,357)	326	24.02 %	107.35 s.	626	46.13 %	110.31 s.	0	0.00 %	0.00 s.	0	0.00 %	0.00 s.
Total (7,039)	**412**	**5.85 %**	**102.51 s.**	**2,684**	**38.13 %**	**74.68 s.**	**2,724**	**38.70 %**	**21.85 s.**	**133**	**1.89 %**	**88.37 s.**

First of all, we have validated the completed versions of Adimen-SUMO using white-box testing techniques [3], and we have not found any inconsistency. Next, we have evaluated the efficiency and competency of each FOL version of Adimen-SUMO. For this purpose, we have used the framework for the evaluation of the competency of SUMO-based ontologies introduced in Álvez *et al.* [6]. The interested reader can find a detailed analysis in Álvez *et al.* [2]. This framework uses *competency questions* (CQs) [18] derived from several predefined question patterns (QPs) and three main knowledge resources: 1) the lexical database WordNet [13], where lexical concepts encoded in *synonym sets* or *synsets* are semantically related by different types of semantic relations such as hyponymy, antonymy, meronymy, etc. 2) a FOL translation of SUMO like Adimen-SUMO and 3) the semantic mapping between WordNet and SUMO [27]. Specifically, our benchmark is composed of 14,324 commonsense CQs obtained from 4 QPs based on hyponymy —2 QPs for nouns and 2 QPs for verbs— and 3 QPs based on antonymy. Each CQ consists of two conjectures: the first is called the *truth-test*, which is expected to be entailed by the ontology and describes the CQ; the second is called *falsity-test*, which is obtained as the negation of the truth-test and is expected not to be entailed by the ontology. Next, we briefly describe our QPs and provide some examples of the resulting CQs. Given a hyponym pair of nouns or verbs, the semantics of the hyponym is subsumed by the semantics of the hyperonym, and our QPs simply state the same property in terms of Adimen-SUMO depending on the mapping relation that is used for connecting the hyponym to SUMO: two options, *instantiation*/*subsumption* or *equivalence*. For example, the synsets *drinking_water* and *beverage* in Figure 1 are connected to *Water* and *Beverage* respectively, and the hyponym (i.e. *drinking_water*) is connected using *subsumption*. Thus, we apply the first QP based on *hyponymy* proposed in [6] and obtain a CQ consisting of the truth-test

$$\exists ?x \; (\; instance(?x, Water) \wedge instance(?x, Beverage \;) \quad (15)$$

and its negation. With respect to antonymy, the semantics of a pair of antonym synsets are incompatible, which is stated by our QPs in terms of Adimen-SUMO depending on the mapping relations that are used for connecting the synsets to SUMO. In this case, there are 3 options: in the two first ones the two synsets are connected using the same relation (either *instantiation*/*subsumption* or *equivalence*), and in the last one the synsets are connected using different relations (one is connected using *instantiation*/*subsumption* and the other one is connected using *equivalence*). For example, the adjectives *liquescent* and *frozen* are connected by *equivalence* to *Melting* and *Freezing* respectively. Thus, we apply the first QP based on *antonymy* proposed in [6] and obtain a CQ consisting of the truth-test

$$\forall ?x \forall ?y \; (\; (\; instance(?x, Melting) \wedge \; instance(?y, Freezing) \;) \; \rightarrow \; \neg ?x = ?y \;) \quad (16)$$

and its negation.

Given a benchmark, two dual tests are performed for each CQ using FOL ATPs: the first test is to check whether, as expected, the truth-test is entailed by the ontology; the second one is to check if the falsity-test is entailed. If ATPs find a proof for either the truth- or the falsity-test, then the CQ is classified as *solved* (or *resolved*). In particular, the CQ is *passing*/*non-passing* if ATPs find a proof for the truth-test/falsity-test. Otherwise (that is, if no proof is found), the CQ is classified as *unresolved* or *unknown*.[8] For example, the CQ described by "*Drinking water is a beverage*", which consists of truth-test (15) and its negation, is classified as *unknown* by using the original version of Adimen-SUMO, as *non-passing* by using Adimen-SUMO augmented by the application of the CWA to *subclass* and *disjoint* assuming disjointness, and as *passing* by using Adimen-SUMO augmented by the application of the CWA to *subclass* and *disjoint* assuming non-disjointness.

Our experimentation has been performed by using Vampire v4.2.2 —which is the *CADE ATP System Competition* (CASC) FOF[9] division winner in 2017 [32, 41] and the latest available stable release[10] of Vampire at the time of our experimentation— in a Intel® Xeon® CPU E5-2640v3@2.60GHz with 2GB of RAM memory per processor. For each test, we have set an execution-time limit of 300 seconds and a memory limit of 2GB.[11] Totally, the experimentation has required almost 300 days/processor of computation effort: 3 ontologies, 14,324 CQs, 2 tests per CQ and 300 seconds per test. All the required knowledge resources —the original ontology Adimen-SUMO and its versions under the CWA, the set of CQs and conjectures, the mapping between SUMO and WordNet v3.0, WordNet v3.0 relation pairs— and the resulting execution reports are available at `https://adimen.si.ehu.es/web/AdimenSUMO`.

We summarize our experimental results in Table 1, where CQs are organized by QP. In the first column (Competency Questions column), we provide the QP type and the number of CQs (between brackets). In the next 9 columns, we provide the number (columns

[8] Given a consistent ontology, ATPs cannot find a proof for both the truth- and the falsity-test.

[9] First-Order Form non-propositional theorems (axioms with a provable conjecture).

[10] `https://vprover.github.io/`

[11] Parameters: `--proof tptp --output_axiom_names on --mode casc -t 300 -m 2048`

#), percentage (columns %) and average runtime (columns T) of CQs that are solved by using each version of Adimen-SUMO: the original version of Adimen-SUMO (OWA, 3 columns), Adimen-SUMO augmented by applying the CWA to *subclass* and *disjoint* assuming disjointness (CWA-D, 3 columns), and Adimen-SUMO augmented by applying the CWA to *subclass* and *disjoint* assuming non-disjointness (CWA-n-D, 3 columns).

From our results, it is easy to see that the CWA surpasses the OWA in our benchmark in terms of competency: the two augmented versions of Adimen-SUMO outperform the original version in terms of solved CQs ($9,781$ and $9,498$ against $7,285$ solved CQs) and that the total number of solved CQs increases more than 50 % ($10,970$ against $7,285$ solved CQs). Further, for each QP the number of solved CQs also increases up to 136 % ($1,502$ against 637 solved CQs from *verb #1*). However, in the case of the CQs that result from *antonym #1*, the original version of Adimen-SUMO outperforms the augmented ones (36 against 29 and 27 solved CQs). This is because when using the augmented versions of Adimen-SUMO the ATP runs out of resources (mainly time) at trying to solve some of the CQs that were already solved by using the original version of the ontology. In total, 359 CQs (4.93 %) that are solved by using the original version of Adimen-SUMO remain unresolved when trying one of the augmented versions of Adimen-SUMO in our experimentation. However, for each QP many new CQs are solved only when using one of the augmented versions of Adimen-SUMO. Moreover, even improving the competency, the augmented versions of Adimen-SUMO also outperforms the original one in terms of efficiency (55.12 s. and 58.75 s. against 61.30 s.), mainly because of the efficiency improvement at solving the CQs obtained from *verb #1* and *antonym #3*. This implies that the new added knowledge has not a deep negative impact in the efficiency of the augmented ontologies. Further, we have checked that the newly added axioms sometimes serve as shortcuts in the proof of problems that were already solved using the original version of Adimen-SUMO.

Additionally, in Table 2 we provide some figures about the CQs that remain unresolved when using the original version of Adimen-SUMO. In the first column (Competency Questions column), we provide the QP from which CQs have been obtained and the number of CQs that remain unresolved when using the original version of Adimen-SUMO (between brackets). The last 12 columns are organized into groups of 3 columns. In each group, we provide the number (# columns), percentage of CQs (% columns) and average runtime (T columns) that are respectively classified as *passing*/*non-passing* by using each augmented version of Adimen-SUMO: Adimen-SUMO augmented by applying the CWA to *subclass* and *disjoint* assuming disjointness (CWA-D), and Adimen-SUMO augmented by applying the CWA to *subclass* and *disjoint* assuming non-disjointness (CWA-n-D).

According to the reported results, the classification of the newly solved CQs strongly depends on the given assumption that we adopt in order to apply the CWA: $2,684$ CQs (38.13 %) are classified as *passing* if assuming non-disjointness, while $2,724$ CQs (38.70 %) are classified as *non-passing* if assuming disjointness. Among them, 732 CQs are solved only when applying the CWA to *subclass* and *disjoint* by assuming disjointness. On the contrary, 527 CQs are solved only when applying the CWA to *subclass* and *disjoint* by assuming non-disjointness. Further, the chosen assumption also influences the kind of CQs that are most frequently solved: the largest amount of CQs obtained from hyponymy are solved when assuming disjointness ($2,773$ against $2,179$ solved CQs obtained from *noun #1*, *noun #2*, *verb #1* and *verb #2*), although the largest number of CQs obtained from antonymy-based QPs are solved when assuming non-disjointness (638 against 363 solved CQs obtained from *antonym #1*, *antonym #2* and *antonym #3*). Regarding average runtimes, it seems that assuming disjointness at applying the CWA to *disjoint* yields to a more efficient augmented version of Adimen-SUMO.

5 Discussion

In this section, we discuss the experimental results reported in the above section.

The experimental results reported in Section 4 can be further improved. First, we think that the ATP runs out of resources (especially time) when trying to prove conjectures that are entailed by some of the augmented versions of Adimen-SUMO. Actually, we have experimentally checked that the ATP runs out of resources using the augmented versions of Adimen-SUMO when trying to solve 359 CQs that are solved by the original version of Adimen-SUMO (less than 5 %). Thus, it is very likely that there are more solvable CQs. Second, we also think that our results are penalized by the poor mapping of adjective synsets as pointed out by [2], since the worst results reported in Table 1 correspond to the CQs obtained from the QPs based on pairs of *antonym* adjectives. Third, we have manually inspected some cases and detected that some knowledge is still under-specified, despite of the application of the CWA. For example, it is not possible to infer from the augmented versions of Adimen-SUMO whether *Animal* and *LinguisticExpression* are *disjoint* or not. Another example of missing knowledge is the axiomatiation of many attributes. We have discovered this problem by analysing the most frequent concepts involved in the *unknown* CQs e.g. *SubjectiveAssessmentAttribute*.

Regarding non-asserted *subclass* pairs, our assumption is that those pairs are not related by *subclass*, since otherwise most of the involved classes would become equal by anti-symmetry as discussed in Section 3. Interestingly, the impact of augmenting Adimen-SUMO by applying the CWA only to *subclass* as described in Subsection 3.1 is really small. On the contrary, the impact of the application of the CWA to *subclass* in combination with the application of the CWA to *disjoint* is much higher. This is especially the case when applying the CWA by assuming disjointness, mainly because the DCA is also applied in the proposal described in Subsection 3.1.

With respect to non-asserted *disjoint* pairs, we have assumed that those pairs are either disjoint (in Subsection 3.2) or non-disjoint (in Subsection 3.3). As described in Section 4, the classification of most of the newly solved CQs depends strongly on the chosen assumption: if assuming disjointness, $2,724$ newly solved CQs are classified as *non-passing*, while $3,136$ newly solved CQs are classified as *passing* when assuming non-disjointness (see Table 2). An example of this is the CQ described in the introduction: "*Drinking water is a beverage*", see truth-test (15). This fact confirms the lack of structural knowledge about classes in Adimen-SUMO.

We have, therefore, proved that structural knowledge is missing and it will be necessary to augment the ontology. We foresee that the classification of the newly solved CQs can guide the application of the CWA to *disjoint* using WordNet as a reliable knowledge source. Let us explain this proposal with two examples. On the one hand, by assuming that *Melting* and *Freezing* are disjoint, the CQ described by "*the adjectives liquescent and frozen are antonym*", consisting of truth-test (16) and its negation, is *passing*, but if non-disjointness is assumed, it is *non-passing*. So, using the knowledge in the ontology and the combination of both augmented versions we can conclude

that *Melting* and *Freezing* must be *disjoint*. On the other hand, the CQ described by "*Drinking water is a beverage*" is *non-passing* when assuming that *Beverage* and *CompoundSubstance* are disjoint but *passing* when assuming non-disjointness. Thus, these classes should be *non-disjoint*. In sum, the combination of assuming disjointness in cases such as *Melting* and *Freezing* and non-disjointness in cases such as *Beverage* and *CompoundSubstance* seems to be appropriate to obtain the correct disjoint/non-disjoint axioms if using WordNet as a reliable knowledge source. In any case, the above described solutions for *disjoint* require the addition of many new axioms. We have proved that the source Adimen-SUMO and its completed versions are comparable in terms of efficiency according to the experimentation reported in Section 4, even though we have added many axioms. However, if we follow suitable structural design criteria, it is not necessary to include additional axioms in a FOL ontology for the application of the CWA to *disjoint*.

One of these possible criteria is linked to the the application of the CWA to *subclass*: Organizing the knowledge around the notion of classes will implicitly provide a solution for *disjoint*. If we define two classes as disjoint iff they do not share any common subclass, the completion of *subclass* itself (see Subsection 3.1) enables deciding whether two classes are disjoint or not. However, at this time, this is not possible because the notion of disjointness in SUMO inappropriately states that two classes are disjoint iff they do not share any common instance

$$\forall ?x_1 \forall ?x_2 \forall ?y \; (\; disjoint(?x_1, ?x_2) \; \rightarrow \\ \neg \; (\; instance(?y, ?x_1) \wedge \\ instance(?y, ?x_2) \;) \;) \tag{17}$$

and not subclasses as we propose:

$$\forall ?x_1 \forall ?x_2 \forall ?y \; (\; disjoint(?x_1, ?x_2) \; \rightarrow \\ \neg \; (\; subclass(?y, ?x_1) \wedge \\ subclass(?y, ?x_2) \;) \;) \tag{18}$$

Further, the inappropriate use of *instance* is extended to most parts of SUMO and makes it possible to infer that many pairs of classes that do not share any common subclass do have common instances. This fact makes it really difficult to correct the axiomatization of *disjoint* in SUMO without reconstructing all the knowledge from almost scratch and, consequently, it prevents the easy application of CWA to *disjoint* in SUMO.

In sum, we consider that the best choice for the practical application of the CWA in a FOL ontology is to follow the suitable design criteria such as the one explained above.

6 Conclusions and Future Work

To the best of our knowledge, up to now the research and evaluation of SUMO-based FOL ontologies have been developed exclusively under the *Open World Assumption*. This paper reports on the first investigation on the application of the *Closed World Assumption* to SUMO-based FOL ontologies. Concretely, we have applied the Careful CWA introduced by [15] to the *subclass* and *disjoint* relations of a FOL version of SUMO. We have checked two CWA formulations for *disjoint*: i) by assuming disjointness and ii) by assuming non-disjointness. We have tested these two formulations on a very large benchmark of 14,324 commonsense competency questions extracted from WordNet and its mapping to SUMO. Summing up, although the size of the ontologies has been increased, the resulting ontologies are far more competent and keep their efficiency. Regardless of the CWA strategy applied to *disjoint*, our research empirically demonstrates that the competency of the ontology can improve more than 50 % when reasoning under the CWA. As a side effect, we have also discovered that the *missing* structural knowledge in SUMO is nowadays one of the main obstacles for its practical application in automated commonsense reasoning tasks. In fact, almost 30 % of the structural knowledge about classes is missing in Adimen-SUMO. Further, our proposal can help ontologists to complete the missing knowledge, for example by using WordNet. Thus, the practical utility of our proposal in tasks that require commonsense reasoning is clear.

Although the approach assuming disjointness obtains the best results, a combination of both approaches should be further investigated. For example, using WordNet as reliable source of knowledge, a possible approach is to weigh each new pair according to the number of solutions in which it is used and its kind (*passing*/*non-passing*), and then proceed to choose the most relevant ones while keeping consistency. Similar approaches could be taken into account by considering other sources of knowledge. Additionally, as discussed, suitable design criteria can facilitate the application of the CWA to FOL ontologies. Future work will focus on implementing the proposed strategies. It will also involve experimenting with other knowledge representation strategies such as the *Unique Name Assumption* (UNA) [23] and testing augmented versions of the ontology with other datasets such us the ones created in the Webchild project [42] and ConceptNet [38].

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their comments, which helped improve this paper considerably.

This work has been partially funded by the the project DeepReading (RTI2018-096846-B-C21) supported by the Ministry of Science, Innovation and Universities of the Spanish Government, and GRAMM (TIN2017-86727-C2-2-R) supported by the Ministry of Economy, Industry and Competitiveness of the Spanish Government, the Basque Project LoRea (GIU18/182), Ixa Group-consolidated group type A by the Basque Government (IT1343-19) and BigKnowledge – *Ayudas Fundación BBVA a Equipos de Investigación Científica 2018*.

REFERENCES

[1] J. Álvez, I. Gonzalez-Dios, and G. Rigau, 'Cross-checking WordNet and SUMO using meronymy', in *Proc. of the 11th Int. Conf. on Language Resources and Evaluation (LREC 2018)*, pp. 4570–4577, (2018).

[2] J. Álvez, I. Gonzalez-Dios, and G. Rigau, 'Commonsense reasoning using WordNet and SUMO: a detailed analysis', in *Proc. of the 10th Global WordNet Conference (GWC 2019)*, pp. 197–205, (2019).

[3] J. Álvez, M. Hermo, P. Lucio, and G. Rigau, 'Automatic white-box testing of first-order logic ontologies', *Journal of Logic and Computation*, **29**(5), 723–751, (2 2019).

[4] J. Álvez, P. Lucio, and G. Rigau, 'Adimen-SUMO: Reengineering an ontology for first-order reasoning', *Int. J. Semantic Web Inf. Syst.*, **8**(4), 80–116, (2012).

[5] J. Álvez, P. Lucio, and G. Rigau, 'Black-box testing of first-order logic ontologies using WordNet', *CoRR*, **abs/1705.10217**, (2017).

[6] J. Álvez, P. Lucio, and G. Rigau, 'A framework for the evaluation of SUMO-based ontologies using WordNet', *IEEE Access*, **7**, 36075–36093, (2019).

[7] G. Antoniou and F. Harmelen, *A Semantic Web Primer (2nd edition)*, MIT Press, 2008.

[8] B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins, 'What are ontologies, and why do we need them?', *IEEE Intelligent Systems*, **14**(1), 20–26, (1999).

[9] K. L. Clark, *Negation As Failure*, 293–322, Springer, Boston, MA, 1978.

[10] M. d'Aquin and N. F. Noy, 'Where to publish and find ontologies? A survey of ontology libraries', *Web Semantics: Science, Services and Agents on the World Wide Web*, **11**, 96–111, (2012).
[11] N. Drummond and R. Shearer, 'The open world assumption', in *eSI Workshop: The Closed World of Databases meets the Open World of the Semantic Web*, volume 15, (2006).
[12] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits, 'Combining answer set programming with description logics for the semantic web', *Artificial Intelligence*, **172**(12), 1495 – 1539, (2008).
[13] *WordNet: An Electronic Lexical Database*, ed., C. Fellbaum, MIT Press, 1998.
[14] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, 'Sweetening ontologies with DOLCE', in *Knowledge Engin. and Knowledge Manag.: Ontologies and the Semantic Web*, ed., A. Gómez-Pérez et al., LNCS 2473, 166–181, Springer, (2002).
[15] M. Gelfond and H. Przymusinska, 'Negation as failure: Careful closure procedure', *Artificial Intelligence*, **30**(3), 273–287, (1986).
[16] M. R. Genesereth, R. E. Fikes, D. Brobow, R. Brachman, T. Gruber, P. Hayes, R. Letsinger, V. Lifschitz, R. Macgregor, J. Mccarthy, P. Norvig, R. Patil, and L. Schubert, 'Knowledge Interchange Format version 3.0 reference manual', Technical Report Logic-92-1, Stanford University, Computer Science Department, Logic Group, (1992).
[17] S. Grimm, P. Hitzler, and A. Abecker, 'Knowledge representation and ontologies', *Semantic Web Services: Concepts, Technologies, and Applications*, 51–105, (2007).
[18] M. Grüninger and M. S. Fox, 'Methodology for the design and evaluation of ontologies', in *Proc. of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI 1995)*, (1995).
[19] K. Hoder and A. Voronkov, 'Sine qua non for large theory reasoning', in *Proc. of the 23rd Int. Conf. on Automated Deduction (CADE-23)*, eds., N. Bjørner and V. Sofronie-Stokkermans, volume 6803 of *Lecture Notes in Computer Science*, 299–314, Springer Berlin Heidelberg, (2011).
[20] I. Horrocks and A. Voronkov, 'Reasoning support for expressive ontology languages using a theorem prover', in *Foundations of Information and Knowledge Systems*, ed., J. Dix et al., LNCS 3861, 201–218, Springer, (2006).
[21] L. Kovács and A. Voronkov, 'First-order theorem proving and Vampire', in *Computer Aided Verification*, eds., N. Sharygina and H. Veith, LNCS 8044, 1–35, Springer, (2013).
[22] A.-A. Krisnadhi, K. Sengupta, and P. Hitzler, 'Local closed world semantics: Keep it simple, stupid!', in *Proc. of the 24th Int. Workshop on Description Logics (DL 2011)*, eds., R. Rosati, S. Rudolph, and M. Zakharyaschev, volume 745 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2011).
[23] V. Lifschitz, L. Morgenstern, and D. Plaisted, 'Knowledge representation and classical logic', in *Handbook of Knowledge Representation*, eds., F. van Harmelen, V. Lifschitz, and B. Porter, 3–88, Elsevier, (2008).
[24] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, 'An introduction to the syntax and content of Cyc', in *Proc. of the Spring Symposium: Formalizing and Compiling Background Knowledge and Its Appl. to Knowledge Repr. and Question Answering*, ed., C. Baral, pp. 44–49. AAAI Press, (2006).
[25] B. Motik and R. Rosati, 'Reconciling description logics and rules', *J. ACM*, **57**(5), 30:1–30:62, (June 2008).
[26] I. Niles and A. Pease, 'Towards a standard upper ontology', in *Proc. of the 2nd Int. Conf. on Formal Ontology in Information Systems (FOIS 2001)*, ed., Guarino N. et al., pp. 2–9. ACM, (2001).
[27] I. Niles and A. Pease, 'Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology', in *Proc. of the IEEE Int. Conf. on Inf. and Knowledge Engin. (IKE 2003)*, ed., H. R. Arabnia, volume 2, pp. 412–416. CSREA Press, (2003).
[28] N. F. Noy and D. L. McGuinness, 'Ontology development 101: A guide to creating your first ontology', Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, (2001).
[29] A. Pease, 'Standard Upper Ontology Knowledge Interchange Format'. Retrieved June 18, 2009, from `http://sigmakee.cvs.sourceforge.net/sigmakee/sigma/suo-kif.pdf`, 2009.
[30] A. Pease and G. Sutcliffe, 'First-order reasoning on a large ontology', in *Proc. of the Workshop on Empirically Successful Automated Reasoning in Large Theories (CADE-21)*, ed., Sutcliffe G. et al., CEUR Workshop Proceedings 257, CEUR-WS.org, (2007).
[31] A. Pease, G. Sutcliffe, N. Siegel, and S. Trac, 'Large theory reasoning with SUMO at CASC', *AI Communications*, **23**(2-3), 137–144, (2010).
[32] F.J. Pelletier, G. Sutcliffe, and C.B. Suttner, 'The Development of CASC', *AI Communications*, **15**(2-3), 79–90, (2002).
[33] D. Ramachandran, R. P. Reagan, and K. Goolsbey, 'First-orderized ResearchCyc: Expressivity and efficiency in a common-sense ontology', in *Papers from the Workshop on Contexts and Ontologies: Theory, Practice and Applications (AAAI 2005)*, ed., P. Shvaiko et al., pp. 33–40. AAAI Press, (2005).
[34] R. Reiter, *On Closed World Data Bases*, 55–76, Springer, Boston, MA, 1978.
[35] R. Reiter, 'A logic for default reasoning', *Artificial intelligence*, **13**(1-2), 81–132, (1980).
[36] S. Schulz, 'E - A brainiac theorem prover', *AI Communications*, **15**(2-3), 111–126, (2002).
[37] S. Siebert, C. Schon, and F. Stolzenburg, 'Commonsense reasoning using theorem proving and machine learning', in *Machine Learning and Knowledge Extraction*, eds., A. Holzinger, P. Kieseberg, A M. Tjoa, and E. Weippl, pp. 395–413, Cham, (2019). Springer International Publishing.
[38] R. Speer, J. Chin, and C. Havasi, 'ConceptNet 5.5: An open multilingual graph of general knowledge', in *Thirty-First AAAI Conference on Artificial Intelligence*, (2017).
[39] S. Staab and R. Studer, *Handbook on Ontologies*, Springer Publishing Company, Incorporated, 2nd edn., 2009.
[40] G. Sutcliffe, 'The TPTP problem library and associated infrastructure', *J. Automated Reasoning*, **43**(4), 337–362, (2009).
[41] G. Sutcliffe and C. Suttner, 'The State of CASC', *AI Communications*, **19**(1), 35–48, (2006).
[42] N. Tandon, G. De Melo, and G. Weikum, 'WebChild 2.0: Fine-grained commonsense knowledge distillation', in *Proc. of ACL 2017, System Demonstrations*, pp. 115–120, (2017).
[43] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A. Minard, A. Palmero Aprosio, and G. Rigau, 'Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news', *Knowledge-Based Systems*, **110**, 60–85, (2016).
[44] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition) - W3C Recommendation 11 December 2012, 2012.