Improving the Competency of First-Order Ontologies

ARTICLE · OCTOBER 2015
DOI: 10.1145/2815833.2815841 · Source: arXiv

CITATION
1

3 AUTHORS, INCLUDING:

Paqui Lucio
Universidad del País Vasco / Euskal Herriko…
40 PUBLICATIONS 101 CITATIONS

German Rigau
Universidad del País Vasco / Euskal Herriko…
158 PUBLICATIONS 1,942 CITATIONS

Available from: Paqui Lucio
Retrieved on: 22 December 2015
Improving the Competency of First-Order Ontologies

Javier Álvez
LoRea Group
University of the Basque Country UPV/EHU
javier.alvez@ehu.eus

Paqui Lucio
LoRea Group
University of the Basque Country UPV/EHU
paqui.lucio@ehu.eus

German Rigau
IXA NLP Group
University of the Basque Country UPV/EHU
german.rigau@ehu.eus

ABSTRACT
We introduce a new framework to evaluate and improve first-order (FO) ontologies using automated theorem provers (ATPs) on the basis of competency questions (CQs). Our framework includes both the adaptation of a methodology for evaluating ontologies to the framework of first-order logic and a new set of non-trivial CQs designed to evaluate FO versions of SUMO, which significantly extends the very small set of CQs proposed in the literature. Most of these new CQs have been automatically generated from a small set of patterns and the mapping of WordNet to SUMO. Applying our framework, we demonstrate that Adimen-SUMO v2.2 outperforms TPTP-SUMO. In addition, using the feedback provided by ATPs we have set an improved version of Adimen-SUMO (v2.4). This new version outperforms the previous ones in terms of competency. For instance, “Humans can reason” is automatically inferred from Adimen-SUMO v2.4, while it is neither deducible from TPTP-SUMO nor Adimen-SUMO v2.2.

Categories and Subject Descriptors
I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods

General Terms
Experimentation

1. INTRODUCTION
Ontologies are being used in a wide range of applications and knowledge based systems [2]. Like any other component of a system, an ontology requires a repetitive process of refinement and evaluation during its development and application lifecycle. Ontologies can be evaluated by considering their use in an application when performing correct predictions on inferencing [25]. In order to enable better reasoning capabilities, the inferencing process should be able to deduce from the ontology as much correct implicit knowledge as possible. In [11], the authors propose to use a set of competency questions (CQs) to evaluate an ontology, which are goals that the ontology is expected to answer. The proposed methodology can be applied to any formal ontology if there exists a decision algorithm for the underlying logic.

In general the process of obtaining CQs is not automatic but creative [8]. Depending on the size and complexity of the ontology, the process of creating a suitable set of CQs is by itself a very challenging and costly task.

Although OWL-DL [14] is currently one of the most common formal knowledge representation formalisms, it is unable to cope with expressive ontologies like Cyc [26], DOLCE [9] or SUMO [21]. Fortunately, state-of-the-art automatic theorem provers (ATPs) for first-order logic (FOL) like Vampire [28] or E [29] are highly sophisticated systems that have been proved to provide advanced reasoning support to substantial FOL conversions of expressive ontologies, including first-ordered Cyc [26], SUMO [15], TPTP-SUMO [24] and Adimen-SUMO [1]. Despite these preliminary experiments, as far as we know, there is no previous work that applies the methodology of [11] to first-order (FO) ontologies using ATPs.

The contributions of this paper are manyfold. First, following [11], we present a new framework to evaluate and improve FO ontologies using ATPs. Second, we introduce a new set of very large and non-trivial CQs designed to evaluate FO versions of SUMO. Our set includes 64 creative CQs for development and more than 7,000 automatically generated CQs for testing. Our set of creative CQs extends the 33 questions from the CSR (Common Sense Reasoning) problem domain of TPTP (Thousands of Problems for Theorem Provers) [30] and the 5 questions described in [1]. Additionally, we have also devised and developed a novel automatic procedure for generating a very large set of non-trivial CQs from a small set of conceptual patterns on the basis of the knowledge encoded in WordNet (WN) [7] and its mapping to SUMO [22]. Third, as a result of the application of our framework, we create a new version of Adimen-SUMO. Finally, using our new framework and CQs, we carry out an empirical comparison of the existing FO versions of SUMO.

According to our experimental results, the new version of Adimen-SUMO outperforms TPTP-SUMO and all previous versions of Adimen-SUMO. For example, from Adimen-SUMO v2.4 ATPs infer that “Tables do not have a brain” and “Humans can reason”, but “Organisms cannot be dead” and “Tables can eat” are not inferred. However, ATPs yield the opposite results from TPTP-SUMO and Adimen-SUMO v2.2 in the four cases. Both the new version of Adimen-

K-CAP 2015, October 07-10, 2015, Palisades, NY, USA
© 2015 ACM. ISBN 978-1-4503-3849-3/15/10$15.00
DOI: http://dx.doi.org/10.1145/2815833.2815841.

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SUMO and the new set of CQs are freely available.Obviously, this type of non-trivial inferences could be very useful for a wide range of knowledge intensive applications. For instance, to help validating the consistency of associated semantic resources like WN, or to derive new explicit knowledge from them. Furthermore, WN is being used world-wide to anchor different types of semantic resources and wordnets in many languages. Therefore, similar inferences can be obtained for other semantic resources and languages other than English. Likewise, WN is connected to several databases such as OpenCyc [27], DBpedia [3, 6] or YAGO [13] thanks to the Linked Open Data (LOD) cloud initiative [5].

In the next two sections, we first introduce SUMO and its FOL versions, and then our adaptation of the methodology proposed by [11]. In Section 4, we illustrate the process of improving an ontology, which yields Adimen-SUMO v2.4, by providing some examples. Next, the process of automatically obtaining a new set of CQs from WN and its mapping to SUMO is described in Section 5. In Section 6, we report on the competency of TPTP-SUMO and the different versions of Adimen-SUMO. In the last two sections, we respectively provide some concluding remarks for discussion and our future research lines.

2. FIRST-ORDER VERSIONS OF SUMO

SUMO has its origins in the nineties, when a group of engineers from the IEEE Standard Upper Ontology Working Group pushed for a formal ontology standard. Their goal was to develop a standard upper ontology to promote data interoperability, information search and retrieval, automated inference and natural language processing.

SUMO is expressed in SUO-KIF (Standard Upper Ontology Knowledge Interchange Format [23]), which is a dialect of KIF (Knowledge Interchange Format [10]). Both KIF and SUO-KIF can be used to write FOL formulas, but its syntax goes beyond FOL. Consequently, SUMO cannot be directly used by FOL ATPs without a suitable transformation [1].

There exist different proposals for converting large portions of SUMO into a FO ontology. In [24], the authors report some preliminary experimental results evaluating the query timeout for different options when translating SUMO into FOL. Evolved versions of the translation described in [24] can be found in the TPTP Library (hereinafter TPTP-SUMO). In [1], we use ATPs for reengineering around 88% of SUMO, obtaining Adimen-SUMO. Both TPTP-SUMO and Adimen-SUMO inherits information from the top and the middle levels of SUMO (from now on, the core of SUMO), thus discarding all the information from the domain ontologies. In Table 1, we provide some figures comparing the explicit content of SUMO, TPTP-SUMO and Adimen-SUMO.

An example of the practical inference capabilities of TPTP-SUMO and Adimen-SUMO is the CQ “Boys are not domestic animals”

(1) 

(\text{instance} \ ?\text{OBJ Boy})

(\text{not}

(\text{instance} \ ?\text{OBJ DomesticAnimal}))

that can be proved using ATPs. In the next sections, we will also provide some CQs that cannot be inferred from neither TPTP-SUMO nor Adimen-SUMO.

3. COMPETENCY QUESTIONS AND FIRST-ORDER ONTOLOGIES

In this section, we describe how to adapt the methodology introduced in [11] for evaluating and improving large and complex FO ontologies using state-of-the-art ATPs like Vampire [28] or E [29].

In [11], the authors propose to evaluate the expressiveness of an ontology by proving completeness theorems w.r.t. a set of CQs. The proof of completeness theorems requires us to check whether a given CQ is entailed by the ontology or not. For this purpose, we use Vampire v3.0, which works by refutation within a given execution-time limit. Similarly, we could also use E or other ATPs that work by refutation. Theoretically, if the conjecture is entailed by the ontology, then ATPs will eventually find a refutation given enough time (and space). However, theorem proving in FOL is a very hard problem, so it is not reasonable to expect ATPs to find a proof for every entailed conjecture [18]. Thus, if the ATP is able to find a proof for a conjecture, then we know for sure that the corresponding CQ is entailed by the ontology. However, if the ATP cannot find a proof, we do not know if (a) the conjecture is not entailed by the ontology or (b) although the conjecture is entailed, the ATP has not been able to find the proof within the provided execution-time limit. Due to the semi-decidability problem of FOL, increasing the execution-time limit is not a solution for con-

\footnote{In this paper, all axioms are to be considered universally closed.}

Table 1: Some figures about SUMO, TPTP-SUMO and Adimen-SUMO

<table>
<thead>
<tr>
<th></th>
<th>SUMO</th>
<th>TPTP-SUMO</th>
<th>Adimen-SUMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>20,081</td>
<td>2,920</td>
<td>1,009</td>
</tr>
<tr>
<td>Classes</td>
<td>5,563</td>
<td>2,086</td>
<td>2,124</td>
</tr>
<tr>
<td>Relations</td>
<td>369</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>Attributes</td>
<td>2,153</td>
<td>68</td>
<td>66</td>
</tr>
<tr>
<td>Total</td>
<td>28,166</td>
<td>5,282</td>
<td>3,407</td>
</tr>
</tbody>
</table>

1\text{http://adimen.si.ehu.es/web/AdimenSUMO}
2\text{http://www.globalwordnet.org/}
3\text{http://www.ontologyportal.org}
4\text{http://www.tptp.org}
5\text{http://www.vprover.org}
jectures that are not entailed. For the same reason, using other systems that do not work by refutation (for example, by model generation) is not a general solution. To overcome this problem, we consider three possibilities when testing the ontology w.r.t. a given CQ: the test may be (i) passing, (ii) non-passing or (iii) unknown, as we next describe.

As proposed in [11], our method is based on a set of CQs written as conjectures in the language of the ontology. The set of CQs is partitioned into two classes: truth-tests and falsity-tests, depending on whether we expect the conjecture to be entailed by the ontology or not. For example, let us consider the CQs “Men cannot be pregnant” and “Organisms cannot be dead”:

\[
(\text{not} \; \text{instance} \; ?\text{HUMAN} \; \text{Pregnant})
\]

As discussed in the above section, we have to improve the ontology when the conjecture corresponding to a truth-test is not proven to be entailed, or when ATPs prove that the conjecture corresponding to a falsity-test is entailed. In the former case, we get no more feedback from ATPs, since no proof is found we do not know whether the conjecture is entailed or not, thus we classify the truth-test as unknown. In the second step, we deal with the set of falsity-tests, which are supposed not to be inferred from the ontology. Hence, we classify a falsity-test as non-passing when the ATP proves that corresponding conjecture is entailed by the ontology, and as unknown when the ATP does not find any proof.

In practice, unknown truth-tests are treated as non-passing, since ATPs have not been able to prove that the corresponding conjectures are entailed by the ontology. On the contrary, unknown falsity-tests are treated as passing. When a test is classified as non-passing (or unknown in the case of truth-tests), we proceed to correct the ontology. The correction is creative and directed by the problem at hand. In the case of truth-tests, we do not obtain any information for the tests that are not classified as passing (since ATPs find no proof), thus the correction is harder and even we do not know if any correction is required. With respect to falsity-tests, the proof generated by ATPs is used to isolate the controversial axioms, performing additional tests when necessary. The feedback provided by ATPs is very helpful for detecting modelling errors. As ATPs are continuously evolving, our method is producing more precise and useful outcomes, but the semi-decidability problem of FOL still remains. In frameworks where the underlying logic is decidable (like OWL-DL), our framework would be also applicable with the advantage of becoming exact.

In order to carry out the experiments described in this paper, we proceed as follows. First, we collect all creative queries available from the literature. In particular, the 33 questions from the CSR (Common Sense Reasoning) problem domain of TPTP (Thousands of Problems for Theorem Provers) [30] and the 5 questions described in [1]. Then, we extend this reduced set of 38 CQs with 26 new creative CQs. All these 64 creative CQs have been classified manually as 50 truth-tests (the 38 old plus 12 new CQs) and 14 falsity-tests (all new). When applying our framework, we use this reduced set of 64 creative CQs as a dataset for development (as explained in Section 4). For improving the competency of the ontology, we only consider the results and traces returned by the ATPs when proving the CQs of this reduced set of creative tests. Second, for testing the competency of new FO versions of SUMO, we automatically create a very large set of CQs derived from WN and its mapping to SUMO (as described in Section 5). In our framework, we use this large set of more than 7,000 automatic CQs as a dataset for testing (see Section 6).

4. IMPROVING A FIRST-ORDER ONTOLOGY

In this section, we report on the experience of improving an FO ontology applying our framework and the set of 64 creative CQs. In particular, we provide some examples of truth- and falsity-tests, explaining how we have used them and the feedback provided by ATPs for improving our ontology. As a result of this process, we have derived Adimen-SUMO v2.4.

As discussed in the above section, we have to improve the ontology when the conjecture corresponding to a truth-test is not proven to be entailed, or when ATPs prove that the conjecture corresponding to a falsity-test is entailed. In the former case, we get no more feedback from ATPs, since no proof is found. Hence, we have to manually check the ontology to search for the modelling error. In the latter case, we obtain a proof, which includes the incorrect axioms. Roughly speaking, the modelling error can refer to ontological concepts (relations or objects) that are used to define the structure and main features of the ontology itself—the so-called basic concepts—or to concepts that serves to describe the knowledge that is contained in the ontology—from now on, non-basic concepts. For example, instance, subclass, disjoint, partition, attribute, subAttribute, contraryAttribute and exhaustiveAttribute are basic concepts in Adimen-SUMO, whereas NullList and Animal are examples of non-basic concepts. According to our experience, there are three typical types of errors than can be informally described as follows:

- Missing characterization: the modelling error refers to a non-basic concept C, but the ontology lacks the axiomatization of C. This is the simplest case, since the solution consists in axiomatizing C.

- Too weak characterization: the modelling error refers to a non-basic concept C that is characterized, but in a way too weak. In this case, the solution consists in repairing the characterization (updating one or more axioms) of C.

- Unsuitable characterization of basic concepts: the modelling error refers to a non-basic concept C which is well-characterized in terms of some basic concept B, but B is not suitable defined. This is the most complex
case, since the solution may require modifying several basic definitions of the ontology, or even its structure.

However, the distinction between basic and non-basic concepts in an ontology is not always clearly stated and is usually dependent upon one’s interpretation. This is the case of SUMO, where all concepts are defined in terms of SUMO itself, preventing a direct translation of SUMO into FOL [1]. This problem also prevents a more formal characterization of modelling errors. Next, we provide some examples of the modelling errors described above and the CQs that have enabled its automated detection by following our proposal.

Regarding truth-tests, let us consider the CQs “A list containing at least one item is not empty (null)” (4), “Tables do not have a brain” (5) and “Men cannot be pregnant” (2):

\[
(\rightarrow\ (\text{List ?LIST}) \ (\text{ITEM Entity}) \ (\text{inList ?ITEM ?LIST})) \ (\text{not} \ (\text{equal ?LIST NullList}))
\]

(5)

\[
(\rightarrow\ (\text{BRAIN Brain}) \ (\text{TABLE Table})) \ (\text{not} \ (\text{properPart ?BRAIN ?TABLE}))
\]

(2)

After some initial experiments with ATPs, we realize that these conjectures are entailed from neither Adimen-SUMO nor TPTP-SUMO. Analysing the knowledge required to answer each question, we find the following problems.

In the case of the CQ “A list containing at least one item is not empty (null)”, the object NullList is only axiomatized to be instance of List, hence a proper characterization is missing. To solve this problem, we include the following axiom in Adimen-SUMO v2.4 as characterization of NullList:

\[
(\text{not} \ (\text{inList ?ITEM NullList}))
\]

(6)

Including this axiom, Adimen-SUMO v2.4 entails “A list containing at least one item is not empty (null)” (4).

The second CQ “Tables do not have a brain” (5) cannot be proved because of a too weak characterization of the concept Animal. In particular, the source of the problem is:

\[
(\rightarrow\ (\text{ATTRIBUTE Dead Living}) \ (\text{subATTRIBUTE Dead Unconscious}) \ (\text{Unconscious ?AGENT SentientAgent}) \ (\text{LIVING ?ATTRIBUTE}))
\]

(9)

We have performed many runs using different ATPs and none of them finds a proof of the goals (9) and (3) from Adimen-SUMO v2.2 or TPTP-SUMO. Likewise, ATPs do not find either any refutation for (9) from Adimen-SUMO v2.4. However, the new characterization of contraryAttribute and exhaustiveAttribute in Adimen-SUMO v2.4 enables ATPs to find a proof of “Organisms cannot be dead” (3). From this proof, we discover that the problem is related with the following axioms:

\[
(\rightarrow\ (\text{ATTRIBUTE Dead Living}) \ (\text{subATTRIBUTE Dead Unconscious}) \ (\text{Unconscious ?AGENT SentientAgent}) \ (\text{LIVING ?ATTRIBUTE}))
\]

(10)

\[
(\rightarrow\ (\text{ATTRIBUTE ConsciousnessAttribute})
\]

(12)

According to axiom (10), it is not possible to have both Living and Dead as attribute. However, by (11) and (12), Dead is an instance of ConsciousnessAttribute. Moreover,
having Dead as attribute implies to also have Living by (13). Analysing this set of ontological axioms, we have decided to remove axiom (11), which defines Dead as subattribute of Unconscious, from Adimen-SUMO v2.4. This incorrectness was hidden in both Adimen-SUMO 2.2 and TPTP-SUMO due to the inappropriate characterization of attributes.

After this development phase, Adimen-SUMO v2.4 passes the 50 creative truth-tests, whereas 23 and 15 creative truth-tests are classified as unknown by TPTP-SUMO and Adimen-SUMO v2.2 respectively. Regarding the creative falsity-tests, all of them are classified as unknown by the three ontologies.

5. DERIVING COMPETENCY QUESTIONS FROM WORDNET

In order to build our benchmark, we have used the mapping from WN to SUMO [22]. This mapping connects each synset of WN to a term of SUMO using three relations: equivalence, subsumption, and instance. These relationships will be denoted by concatenating the symbols ‘=’ (equivalence), ‘⊂’ (subsumption) and ‘@’ (instance) to the corresponding SUMO concept. For example, pilot$	ext{2}_3$, education$	ext{2}_3$, and zero$	ext{2}_3$ are connected to Pilot$\equiv$, EducationalProcess+ and Integer@. Additionally, the complementary of the relations equivalence and subsumption are also used.

The mapping from WN to SUMO uses terms from the core of SUMO, but also from the domain ontologies. However, both TPTP-SUMO and Adimen-SUMO only use axioms from the core of SUMO. Thus, our first task has been to obtain a mapping from WN to the core of SUMO on the basis of the mapping from WN to SUMO. To this end, for each WN synset not mapped to a term covered by both TPTP-SUMO and Adimen-SUMO, we have conveniently used the structural relations of SUMO (instance, subclass, subrelation and subAttribute) to inherit the term of the core of SUMO to which the synset is connected. Note that this process sometimes requires to modify the mapping relation. For example, the synset frying$	ext{1}_3$ is connected to the SUMO class Frying$=\equiv$, which belongs to the domain ontology Food. In the same domain ontology, Frying$=\equiv$ is defined to be subclass of Cooking, which is defined in the top level of SUMO. Consequently, Frying$=\equiv$ is not defined in the core of SUMO, but Cooking is. Thus, the synset frying$	ext{1}_3$ can be connected to Cooking in the resulting mapping. However, instead of equivalence, frying$	ext{1}_3$ is connected to Cooking by the subsumption mapping relation: that is, Cooking$\subset$. The total number of mappings to the core of SUMO (114,948) is slightly smaller than the number of mappings to SUMO (115,872) since some terms are not properly defined. This is mainly due to the fact that some terms in the mapping derived from older versions of SUMO are not longer available in the current one. For example, the synsets salmon$	ext{1}_3$ and architect$	ext{1}_3$ are respectively connected to the SUMO concepts Salmon$=\equiv$ and Architect$=\equiv$, which do not appear in the latest versions of SUMO.

After obtaining a suitable mapping from WN to the ontologies that we want to compare, we have designed several conceptual patterns of questions regarding the information about antonyms and processes in WN.

5.1 Antonym patterns

WN provides a set of 8,689 antonym-pairs, including nouns, verbs, adjectives and adverbs, from which 7,410 antonym-pairs can be properly mapped to the core of SUMO, as described above. However, we only consider the antonym-pairs where both synsets are connected using the equivalence mapping relation (in total, 190 pairs), discarding those where the subsumption mapping relation is used. For these 190 antonym-pairs, we propose two conceptual patterns of questions. The first pattern is based on the fact that two SUMO classes connected to antonym synsets of WN cannot have common instances. For example, the antonym synsets frozen$	ext{1}_3$ and liqueuous$	ext{1}_3$ are respectively connected to Freezing$\equiv$ and Melting$\equiv$. Thus, from the above antonym-pair, we derive the following competency question:

\[
\begin{align*}
&\text{(not exists (}\exists x\text{) Freezing(}\text{x}\text{))} \\
&\quad \text{and} \quad \text{(instance } \exists x \text{ Melting(}\text{x}\text{))}
\end{align*}
\]

Similarly, the second conceptual pattern states that two attributes connected to antonym synsets are not compatible. For example, from the antonym synsets waking$	ext{1}_3$ and sleeping$	ext{1}_3$, which are connected to Awake$\equiv$ and Asleep$\equiv$, we derive the following competency question:

\[
\begin{align*}
&\text{(not exists (}\exists x\text{) Awake(}\text{x}\text{))} \\
&\quad \text{and} \quad \text{(attribute } \exists x \text{ Awake(}\text{x}\text{))}
\end{align*}
\]

Applying these two patterns on the 190 antonym-pairs where both synsets are connected used equivalence, we obtain 64 different truth-tests. By negating each of the above 64 CQs, we also obtain 64 different falsity-tests.

5.2 Process patterns

Regarding processes, we have used the information in the morphosemantic database, which contains semantic relations between morphologically related nouns and verbs. From the 14 semantic relations defined in the database, we select agent, result, instrument and event. The first three ones relate a process (verb) which its corresponding agent / result / instrument (noun), from which we infer 1,280 CQs by simply stating the same property in terms of SUMO. For example, WN establishes that the result of compose$	ext{1}_3$ is a composition$	ext{1}_3$, which are respectively mapped to ComposingMusic$+\equiv$ and MusicalComposition$=\equiv$. Thus, if the noun and the verb are mapped to different SUMO class constants, our hypothesis is that the mapping is wrong. Following this criterion,

\[
\begin{align*}
&\text{(exists (}\exists x \text{) ComposingMusic(}\text{x}\text{))} \\
&\quad \text{and} \quad \text{(result } \exists x \text{ MusicalComposition(}\text{x}\text{))}
\end{align*}
\]

As before, we also obtain 1,280 falsity-tests by negating the previous ones.

The last relation event connects nouns and verbs referring to the same process. Being the same process, we assume that both the noun and the verb should be mapped to the same class of SUMO. Thus, if the noun and the verb are mapped to different SUMO class constants, our hypothesis is that the mapping is wrong. Following this criterion,
we propose different conceptual patterns of questions, depending on the used mapping relations, with the purpose of detecting wrong-mappings. If both synsets are connected to two different SUMO class constants using the equivalence mapping relation, it should be possible to prove that the two class constants denote different classes. For example, kill\textsubscript{10} and killing\textsubscript{2} are respectively connected to the SUMO classes Death= and Killing=, hence we derive the following CQ:
\[
\text{(not)}
\]
\[(\text{equal Death Killing})]\n
Using this pattern, we derive 25 CQs. The second pattern of question focuses on the case where the synsets are connected to different SUMO class constants and using different mapping relations. That is, one synset is connected using the equivalence mapping relation, whereas the other synset is connected using subsumption. Being the mapping information less precise than in the first case, it does not suffice to prove that the classes are different. In this case, the pattern states that the class connected using equivalence cannot be subclass of the class connected using subsumption. For example, event relates repair\textsubscript{1}, which is connected to Repairing=, and repair\textsubscript{2}, which is connected to Pretending+. Therefore, we assume that Repairing cannot be subclass of Pretending, deriving the following CQ:
\[
\text{(not)}
\]
\[(\text{subclass Repairing Pretending})]\n
From the second event pattern of questions, we derive 330 CQs. In fact, this second pattern can be seen as a particular case of the third one, where both synsets are connected using the subsumption mapping relation. In this case, the pattern states that none of the connected SUMO classes can be subclass of the other one. For example, event relates the synsets measure\textsubscript{1} and appraisal\textsubscript{2}, which are respectively connected to Judging+ and Comparing+. Consequently, we derive the following CQ:
\[
\text{(not)}
\]
\[(\text{subclass Judging Comparing})
\]
\[(\text{subclass Comparing Judging})]\n
Using this third pattern, we obtain 1,857 CQs.

In total, we obtain 2,212 truth-tests by stating that the mapping is correct, and the corresponding 2,212 falsity-tests stating that the mapping is correct.

### 6. Evaluating First-Order Ontologies

In this section, we summarize the evaluation results of TPTP-SUMO and the different versions of Adimen-SUMO using the methodology proposed in Section 3. For this evaluation, we have used the set of 7,112 CQs that have been automatically obtained from WN, as described in the above section.

Table 2 sums up some runtime figures of the ATP Vampire 3.0\textsuperscript{8} [28] when evaluating TPTP-SUMO and Adimen-SUMO with an execution time limit of 600 seconds.\textsuperscript{9} For each ontology, we provide the number of passing (P column), non-passing (N column) and unknown CQs (U column), together with the average runtime of passing (t(P) column) and non-passing (t(N) column) CQs. It is worth to remark that the average runtime of the CQs classified as unknown is the maximum execution time (600 seconds), since no proof is found. From the results, it is clear that Adimen-SUMO v2.4 outperforms Adimen-SUMO v2.2 in terms of competency in both the truth-test (more passing tests) and the falsity-test category (less non-passing tests). Regarding efficiency, the average runtime of Adimen-SUMO v2.4 is longer since the passed additional tests require more complex proofs. Similarly, Adimen-SUMO v2.2 outperforms TPTP-SUMO in the truth-test category, since TPTP-SUMO only passes 4 truth-tests while Adimen-SUMO v2.2 passes 89. Regarding falsity-tests, TPTP-SUMO is the ontology with less non-passing tests, but the average runtime is clearly longer. Thus, we think that the number of non-passing falsity-tests of TPTP-SUMO would be larger if we used a longer execution time limit for the experimentation.

Recall that non-passing falsity-tests can provide useful information to improve the ontology, as in the case of the CQ “Organisms cannot be dead” (see Section 4). Additionally, non-passing falsity-tests also provide very useful information. For example, the verbs whisper\textsubscript{1} and shout\textsubscript{2}, which are antonyms in WN, are mapped to the SUMO classes Speaking= and Vocalizing= respectively. Being antonyms, we expect that these two SUMO classes, Speaking and Vocalizing, do not have any common instance, as stated by the next conjecture which corresponds to a CQ included in the

\textsuperscript{8}http://www.vprover.org

\textsuperscript{9}In this experimentation, we have used a standard 64-bit Intel® Core™ i7-2600 CPU @ 3.40GHz desktop machine with 16GB of RAM.
automatic falsity-test set:

\[
(\exists X) (\text{exists} (\exists X) (\text{and} (\text{instance} ?X \text{ Vocalizing}) (\text{instance} ?X \text{ Speaking})))
\]  

However, ATPs can infer the above conjecture from Adimen-SUMO v2.4. That is, Adimen-SUMO v2.4 does not pass this falsity-test. This fact serves to detect that the mapping of the verbs \(\text{whisper}_1^v\) and \(\text{shout}_1^v\) to SUMO is not suitable. On the contrary, the CQ is classified as unknown when evaluating TPTP-SUMO and Adimen-SUMO v2.2, which prevents to detect the incorrect mapping.

7. DISCUSSION

We have shown a new framework and experimental results for evaluating and improving large and complex FO ontologies using ATPs. Our results show the appropriateness of using ATPs in debugging FO ontologies, as well as their practical use for inferring non-trivial statements.

Although the authors of [4] claim for the necessity of proving the formal faithfulness of a logical translation of SUMO, the formats KIF and SUO-KIF lack a formal notion of logical consequence and a formal deduction system,\(^{10}\) where that kind of mathematical results (such as conservative extension) relies on. Thus, following [11], our repairs just intend to improve the reasoning capabilities of the ontology.

Another possible topic of discussion is the need for some clear quality criteria of the CQs. For instance, all our CQs are universally closed formulas of the form

\[
(P_1 \land P_2 \land \ldots \land P_n) \rightarrow P
\]  

where \(P\) is not deducible from \(\{P_1, P_2, \ldots, P_n\}\) without the help of the knowledge contained in the ontology. That is, none of our CQs includes information in the premises that should be inferred from the ontology. Note that this is not the case of some of the tests from the CSR problem domain of TPTP [30]. Thus, our set of CQs includes a clean version of these goals all the properties \(P_i\) in (21) that should be deducible from the ontology have been removed.

Finally, our set of CQs do not only have conjectures that are expected to be deducible, but it also contains conjectures expected not to be deducible. Obviously, this is also a very interesting experimentation, since these conjectures allow the detection of modelling errors when proved.

8. FUTURE WORK

The new framework presented in this paper opens multiple avenues for future research. Now, we are working in order to improve the competency of Adimen-SUMO. In the case of non-passing automatic tests, this implies to correct either a) the ontology itself, b) some mappings from WN to the ontology, or c) some WN relations. In parallel, we also want to enlarge our current set of CQs by gathering more questions from WN and its mapping to SUMO or alternative datasets. Additionally, it would be very interesting to determine which parts of the ontology are used to solve a set of CQs. Thus, in order to further debug Adimen-SUMO, we are also exploring the possibility to automatically derive an exhaustive set of questions from its axiomatization.

\(^{10}\)A formal declarative semantics for the FO sublanguage SKIF of KIF is given in [12].

Our framework does not only allow to measure the competency of different SUMO-based ontologies, but also its efficiency when solving a large set of non-trivial inferences. In this sense, we are investigating more efficient representations of the ontology. Additionally, our framework can act as a new benchmark for testing the performance of FOL ATPs.

Another open research line will focus on proving the consistency of Adimen-SUMO. For instance, on the basis of a modular approach [19] or by including into our framework sophisticated model finders of the type of iProver [17].

Finally, we plan to develop automatic procedures to exploit Adimen-SUMO and its complete mapping to WN for automatically inferring new semantic properties and relations between WN concepts, or validating the consistency of resources associated to WN such as Cyc, DBpedia or Yago.

9. ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their insightful comments. This work has been partially funded by the Spanish Projects SKaTer (TIN2012-38584-C06-02) and COMMAS (TIN2013-46181-C2-2-R), the Basque Project LoRea (GIU12/26) and grant BAILab (UFI11/45).

10. REFERENCES


