Improving the Competency of First-Order Ontologies

Javier Álvez Paqui Lucio German Rigau

eman ta zabal zazu

UPV EHU

University of the Basque Country

LoRea & IXA NLP Groups

K-Cap 2015 – The 8$^{\text{th}}$ International Conference on Knowledge Capture
October 7-10, 2015, Palisades, NY, USA

Funded by SKaTer (TIN2012-38584-C06-02), COMMAS (TIN2013-46181-C2-2-R) and LoRea (GIU12/26)

Development of First-Order Ontologies

- Our research focuses on first-order ontologies (eg. SUMO)
- Its development requires an iterative and manual process of refinement and evaluation [1]
- For its evaluation, one may consider their use in applications when performing correct predictions
    - Very small data-sets are available (38 conjectures)

Evaluation of Ontologies

- Grüninger & Fox proposed a methodology for the evaluation of ontologies [3]
- The methodology is based on *Competency Questions* (CQs):
  - Goals that the ontology is expected to answer
- Obtaining CQs is not automatic but creative [2]
- Creating a suitable set of CQs is a very challenging and costly task
- This methodology has not been previously applied using first-order logic (FOL) automatic theorem provers (ATPs)

Our Contributions

- A new framework to evaluate and improve the competency of first-order (FO) ontologies using ATPs
- A new set of very large and non-trivial CQs:
  - 64 *creative* tests, including the 33 CQs from the CSR (Common Sense Reasoning) problem domain of TPTP (Thousands of Problems for Theorem Provers) and the 5 CQs from [1]
  - 7,112 *automatic* tests, obtained from a small set of conceptual patterns on the basis of the knowledge in WordNet and its mapping to SUMO
- An improved version of Adimen-SUMO (v2.4)

Outline

SUMO

- *Suggested Upper Merged Ontology*
- Pushed by the *IEEE Standard Upper Ontology* Working Group
- Its goal is to promote data interoperatibility, information search and retrieval, automated inference and natural language processing
- SUMO syntax goes beyond FOL

First-Order Versions Of SUMO

- Two different proposals:
  - TPTP-SUMO [4], which can be found in the TPTP Library
  - Adimen-SUMO [1], which can be found in http://adimen.si.ehu.es/web/AdimenSUMO
- Those ontologies only inherit information from the top and the middle levels of SUMO
- Some figures:

	SUMO	TPTP-SUMO	Adimen-SUMO
Objects	20,081	2,920	1,009
Classes	5,563	2,086	2,124
Relations	369	208	208
Attributes	2,153	68	66
Total	**28,166**	**5,282**	**3,407**

Using FOL ATPs

- Vampire v3.0 (and other FOL ATPs) works by refutation within an execution-time limit
- The methodology proposed by Grüninger & Fox consists in proving *completeness theorems*:
  - Checking whether a CQ is entailed by the ontology *or not*
- Theoretically, if the conjecture is entailed, ATPs will find a refutation
- But ATPs do not find a refutation for every entailed conjecture:
  - If ATPs find a proof, it is sure that the CQ is entailed
  - If not, there are two possibilities:
    - The CQ is not entailed
    - The CQ is entailed, but ATPs cannot find a proof within the execution-time limit

Evaluation (I)

- The set of CQs is partitioned into two classes:
  - *Truth-tests*: expected to be entailed

```
( =>
    (and
        (instance ?HUMAN Human)
        (attribute ?HUMAN Pregnant))
    (not
        (instance ?HUMAN Man)))
```

  - *Falsity-tests*: expected not to be entailed

```
(=>
    (instance ?ORG Organism)
    (not
        (attribute ?ORG Dead)))
```

Evaluation (II)

- Tests may be classified as:

(A) *Passing*
(B) *Non-passing*
(C) *Unknown*

- The method proceeds in two steps:
  - First step – Truth-tests
    - If ATPs find a proof, the test is classified as *passing*
    - Otherwise, the test is classified as *unknown*
  - Second step – Falsity-tests
    - If ATPs find a proof, the test is classified as *non-passing*
    - Otherwise, the test is classified as *unknown*

Improvement

- Two cases:
  - Non-passing falsity-tests:
    - The proof provided by ATPs includes the incorrect axioms
  - Unknown truth-tests:
    - Increase the execution-time limit
    - Manually checking the ontology with the help of ATPs
      - Decomposing the conjecture into several subgoals and try to prove the subgoals by separate
      - Picking by hand the axioms in the ontology that should enable the proof
- Typical problems:
  - Undefined concepts
  - Incomplete definition of properties
  - Unsuitable characterization of meta-concepts

The Mapping from WordNet to SUMO

- Each synset of WordNet is connected into a SUMO concept using 3 relations (and its complementaries):

$=$	Equivalence
$+$	Subsumption
$@$	Instance

- The mapping uses the top and middle level of SUMO, but also the domain ontologies:

$education^4_n$	$\mapsto$	$EducationalProcess+$	(Top level)
$zero^1_a$	$\mapsto$	$Integer@$	(Top level)
$frying^1_n$	$\mapsto$	$Frying=$	(*Food* ontology)

- Adimen-SUMO (and TPTP-SUMO) only inherits information from the top and middle levels of SUMO

Inheriting a Mapping from WordNet to Adimen-SUMO

- On the basis the structural relations of SUMO:

instance
subclass
subrelation
subAttribute

- For example:

		Cooking+	(Top level)
		↑	
$frying^1_n$	$\mapsto$	*Frying*=	(*Food* ontology)

Automatically Obtaining CQs

- Different conceptual patterns based on:
  - Antonym-pairs provided by WordNet:

    $frozen_n^1$ vs. $liquescent_n^1$

  - The morphosemantic database of WordNet, which contains semantics relations between morphologically related nouns and verbs
    - *agent*, *result* and *instrument*

      The *result* of $compose_v^2$ is a $composition_n^4$

    - *event*

      $kill_v^{10}$ and $killing_n^2$ denote the same *event*

Antonym Patterns

- WordNet provides 8,689 antonym-pairs
  - In 190 antonym-pairs, both synsets are connected using *equivalence*
- Two conceptual patterns, focusing on classes and attributes
- We obtain 64 truth-tests
  - By negation, we also obtain 64 falsity-tests

Antonym Patterns: Classes

- Two SUMO classes connected to antonym synsets of WordNet cannot have common instances
- Example:
  - $frozen^1_n$ and $liquescent^1_n$ are antonym:

$$\begin{array}{ccc} frozen^1_n & \mapsto & Freezing{=} \\ liquescent^1_n & \mapsto & Melting{=} \end{array}$$

  - Proposed truth-test:

```
(not
    (exists (?X)
        (and
            (instance ?X Freezing)
            (instance ?X Melting))))
```

Antonym Patterns: Attributes

- Two SUMO attributes connected to antonym synsets of WordNet are not compatible
- Example:
  - $waking_n^1$ and $sleeping_n^1$ are antonym:

$$\begin{array}{rcl} waking_n^1 & \mapsto & Awake= \\ sleeping_n^1 & \mapsto & Asleep= \end{array}$$

  - Proposed truth-test:

```
(not
    (exists (?X)
        (and
            (attribute ?X Awake)
            (attribute ?X Asleep))))
```

Relation Patterns: *agent*, *result*, *instrument*

- *agent*, *result* and *instrument* relate a process (verb) with its corresponding agent / result / instrument (noun)
- We obtain 1,280 truth-tests by stating the same property in terms of SUMO
  - By negation, we also obtain 1,280 falsity-tests
- Example:
  - The *result* of $compose_v^2$ is a $composition_n^4$:

$compose_v^2$	$\mapsto$	$ComposingMusic+$
$composition_n^4$	$\mapsto$	$MusicalComposition=$

  - Proposed truth-test:

```
(exists (?X ?Y)
    (and
        (instance ?X ComposingMusic)
        (result ?X ?Y)
        (instance ?Y MusicalComposition)))
```

Relation Patterns: *event*

- *event* connects nouns and verbs referring to the same process
- Being the same process, the noun and the verb should be mapped to the same SUMO class
  - If not, we suppose that the mapping is wrong
- From 3 conceptual patterns depending on the mapping relations, we obtain 2,212 truth-tests/falsity-tests by stating that the mapping is wrong/correct
- Example:
  - $kill_v^{10}$ and $killing_n^2$ are related by *event*:

$$\begin{array}{ccc} kill_v^{10} & \mapsto & Death= \\ killing_n^2 & \mapsto & Killing= \end{array}$$

  - Proposed truth-test:

```
(not
    (equal Death Killing))
```

Improving Adimen-SUMO

- We have applied our framework to Adimen-SUMO v2.2
- We have used the set of 64 creative tests as a dataset for development
  - 50 truth-tests (12 new)
  - 14 falsity-tests (all new)
- Summary:
  - 15 truth-tests were classified as *unknown*
  - 1 falsity-test was classified as *non-passing*
- As result, we have obtained Adimen-SUMO v2.4

Evaluating the Competency of Adimen-SUMO

- We have evaluated the competency of TPTP-SUMO, Adimen-SUMO v2.2 and Adimen-SUMO v2.4
- Vampire v3.0 (execution-time limit: 600 seconds)

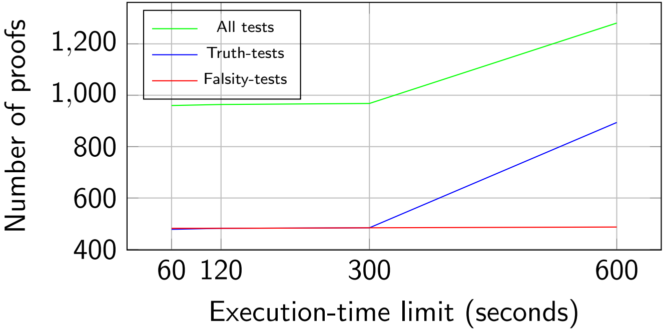
	TPTP-SUMO	Adimen-SUMO v2.2	Adimen-SUMO v2.4
Truth-tests	Passing	Passing	Passing
Antonym pattern (64)	3	17	45
Relation pattern (1,280)	0	11	176
Event pattern #1 (25)	0	2	7
Event pattern #2 (330)	0	26	115
Event pattern #3 (1,857)	1	33	551
Total (3,556)	**4**	**89**	**894**
Falsity-tests	Non-passing	Non-passing	Non-passing
Antonym pattern (64)	4	2	5
Relation pattern (1,280)	4	31	22
Event pattern #1 (25)	0	0	0
Event pattern #2 (330)	71	72	72
Event pattern #3 (1,857)	387	388	388
Total (3,556)	**466**	**493**	**487**

Evaluating the Competency of Adimen-SUMO: Summary

- Adimen-SUMO v2.4 clearly outperforms Adimen-SUMO v2.2 and TPTP-SUMO in the truth-test category
- The results in the falsity-test category are quite similar
- Non-passing and unknown tests may be due to:
  - The mapping
  - WordNet relations
  - The ontology itself
- Some CQ may be unsuitable

Evaluating the Efficiency of Adimen-SUMO

- We have also evaluated the efficiency of Adimen-SUMO v2.4
- In particular:
  - More and more complex truth-tests are solved as the execution-time limit becomes longer
  - On the contrary, the number of non-passing falsity-tests does not substantially increases



- These results will be presented in the poster session

Conclusions and Ongoing Work (I)

- Using our framework, we have successfully evaluated and improved the competency of Adimen-SUMO
- Additionally:
  - Our framework also enables to measure the efficiency of ontologies when solving CQs
  - Our framework can act as a new benchmark for testing the performance of FOL ATPs
- Adimen-SUMO, our benchmark dataset of 7,112 CQs and execution reports are freely available:

  http://adimen.si.ehu.es/web/AdimenSUMO

Conclusions and Ongoing Work (II)

- We are correcting:
  - Adimen-SUMO
  - Some mappings from WordNet to SUMO
  - Some WordNet relations
- We are improving and enlarging our current set of CQs
- We also plan to automatically exploit Adimen-SUMO and the mapping to WordNet:
  - Inferring new semantic relations between WordNet concepts
  - Validating the consistency of resources such as Cyc, DBpedia or Yago

References

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.

M. Fernández-López, A. Gómez-Pérez, and M. C. Suárez-Figueroa.
Methodological guidelines for reusing general ontologies.
Data & Knowledge Engineering, 86:242–275, 2013.

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.

A. Pease and G. Sutcliffe.
First-order reasoning on a large ontology.
In Sutcliffe G. et al., editor, *Proc. of the Workshop on Empirically Successful Automated Reasoning in Large Theories (CADE-21)*, CEUR Workshop Proceedings 257. CEUR-WS.org, 2007.