Mapping Natural Language to Description Logic

Bikash Gyawali Anastasia Shimorina Claire Gardent Samuel Cruz-Lara Mariem Mahfoudh

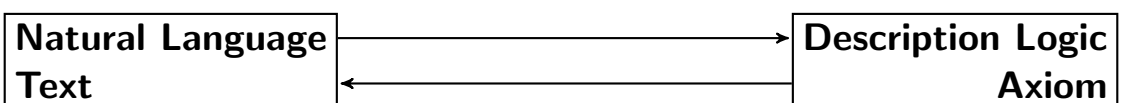
CNRS/LORIA, Nancy, France

May 31, 2017

Introduction

A reversible approach that :

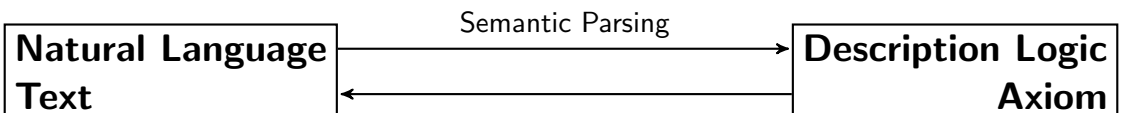
- Maps Natural Language (NL) Sentences to Description Logic (DL) Axioms
- Generates Text to describe DL Axioms.



Introduction

A reversible approach that :

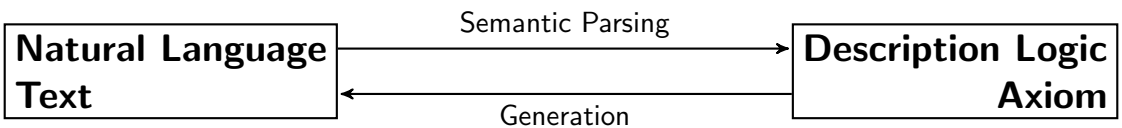
- Maps Natural Language (NL) Sentences to Description Logic (DL) Axioms
- Generates Text to describe DL Axioms.



Introduction

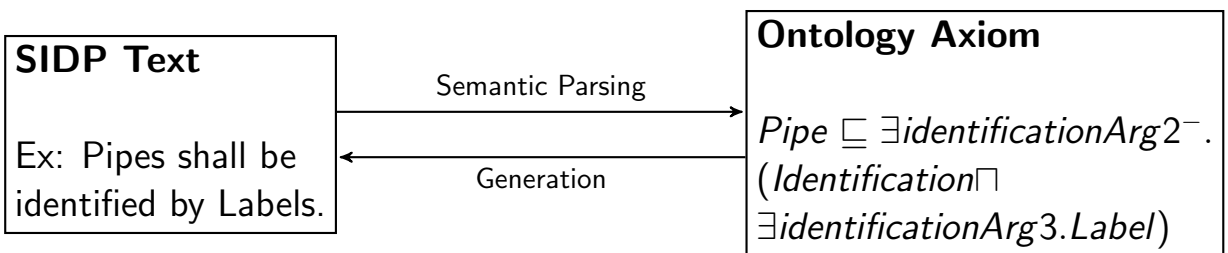
A reversible approach that :

- Maps Natural Language (NL) Sentences to Description Logic (DL) Axioms
- Generates Text to describe DL Axioms.



Context

- Map System Installation Design Principle (SIDP) text to Ontology Axioms.
- Airbus Industry.



Motivation

From Text to Model (Semantic Parsing) :

- Semantic Reasoning on text.
- Knowledge Base (KB) enrichment.
- Manual mapping is difficult : time-consuming, expertise needed.
- Text keeps evolving : Consistency of newly updated SIDPs with existing ones.

From Model to Text (Generation) :

- Easy comprehension of complex axioms.
- Verification of Parsing results.

Outline of this Talk

- Related Work
- Contributions
- Approach Overview – Resources and Methodologies
- Experiments – Results and Evaluation
- Conclusion

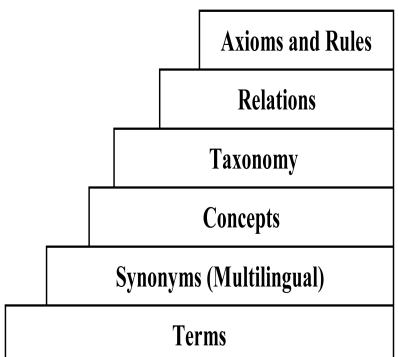
Related Work

Semantic Parsing :

- Currant et al. 2007, McCartney and Manning 2007 : First Order Logic.
- Ge and Mooney 2009, Berant et al., 2013, Bordes et al. 2014, Wang et al. 2015 : Require Parallel text-data corpus to learn.

Ontology Learning :

- Mädche and Staab 2000, Volker et al. 2007, Tablan et al. 2006, Zouaq and Nkambou 2008
  - Identify new concepts, instances and taxonomy of concepts.
  - Identify new properties and their values for instances.
  - No processing of sentence level axioms.



Related Work

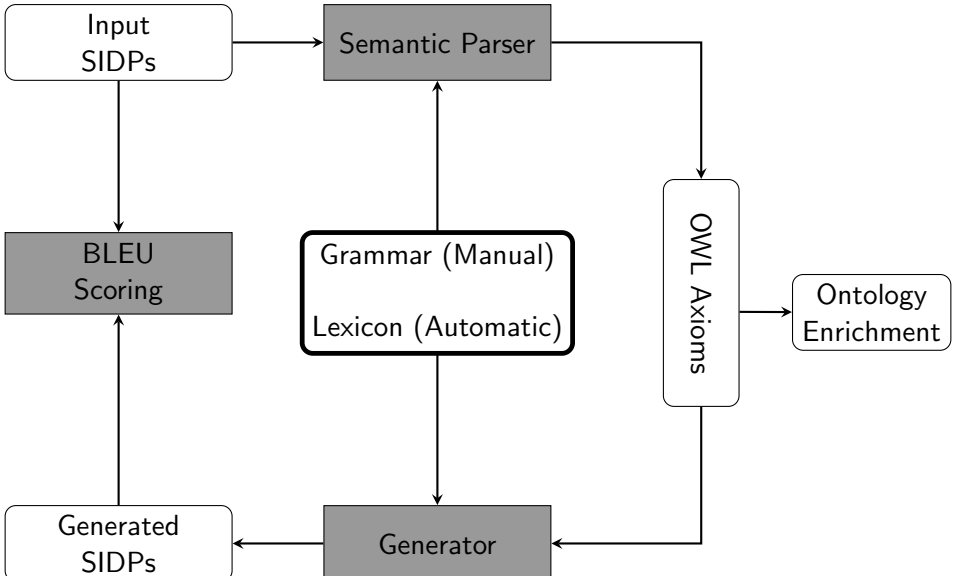
Generation :

- Dimitrios et al. 2007, Androtsopoulos et al. 2013, Power et al. 2010 : Large Hand-written modules.
- Belz 2008, Angeli et al. 2010, Konstas and Lapata 2012 : Require Parallel data-text corpus to learn.
- Duma et al. 2010, Blake et al. 2013, Schilder et al. 2013 : Generation from set of RDF triples.

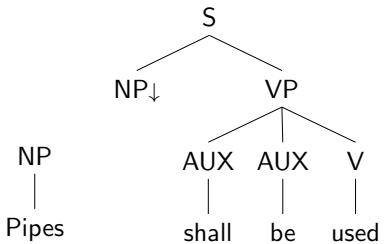
Contributions

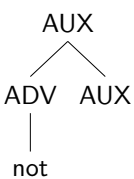
- Derive complex DL **Axioms** from Natural Language **Sentences**.
- Regeneration as a measure of Semantic Parse accuracy.
- Ontology Enrichment using derived Axioms
- Reversible (Semantic Parser – Generator) and Robust Framework.

Approach Overview



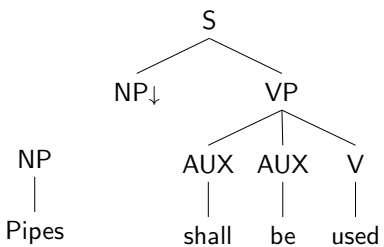
Handcrafted Grammar : FB-LTAG with Unification Based Semantics.

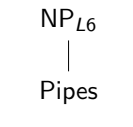


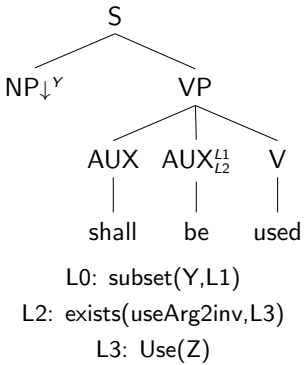


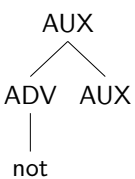
Resources (Grammar and Lexicon)

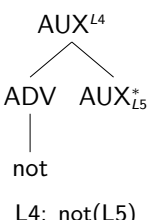
Handcrafted Grammar : FB-LTAG with Unification Based Semantics.











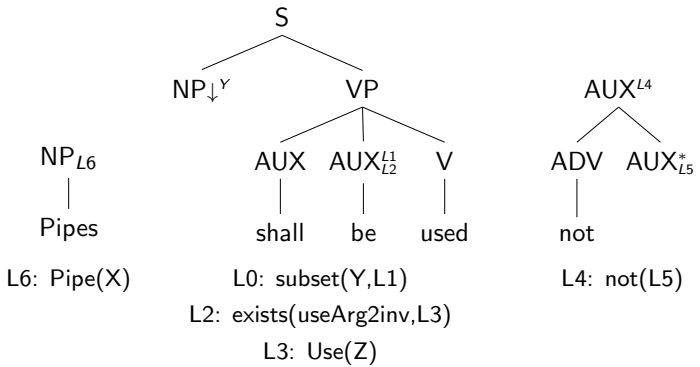
Parsing and Generation

Three main steps

1. Select grammar trees based on input (words or semantic literals)

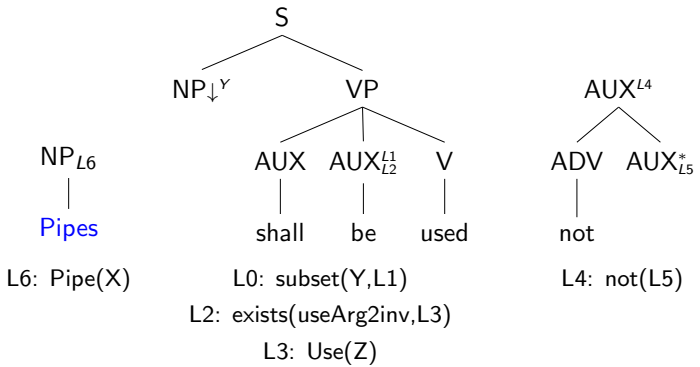
2. Combine selected trees using adjunction and substitution

3. Extract solutions (semantic representations or sentence)



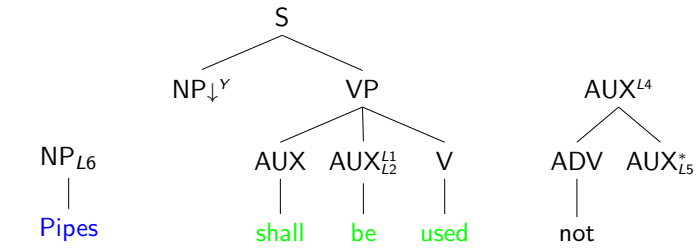
Parsing :

Pipes shall not be used.
Pipes shall be used.



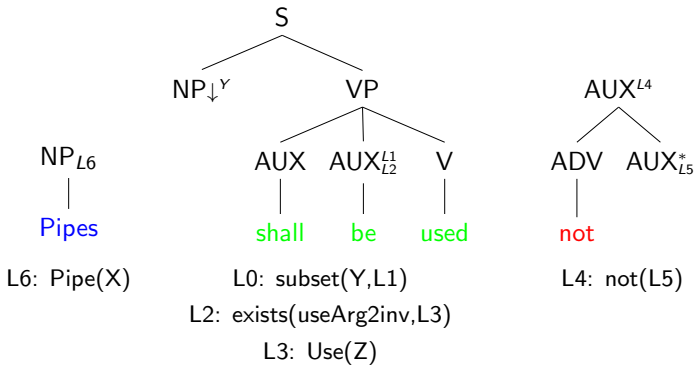
Parsing :

Pipes shall not be used. L6:Pipe(X)
Pipes shall be used.



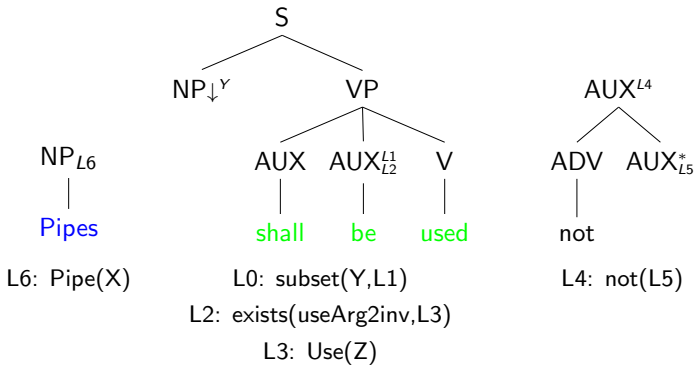
Parsing :

Pipes shall not be used. L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall be used.



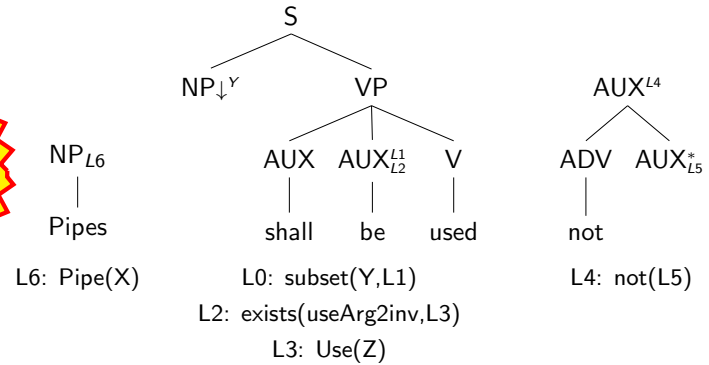
Parsing :

Pipes shall not be used. L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.



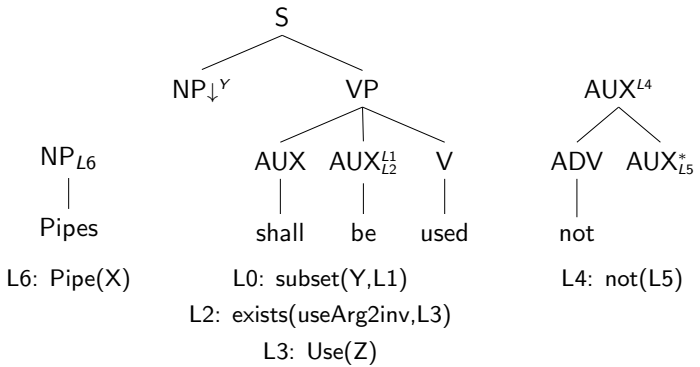
Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)



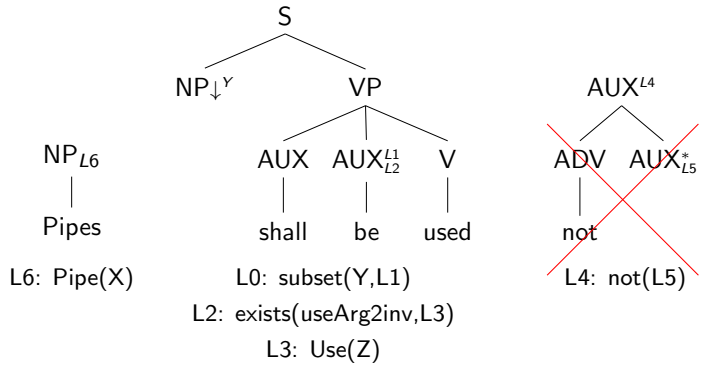
Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)



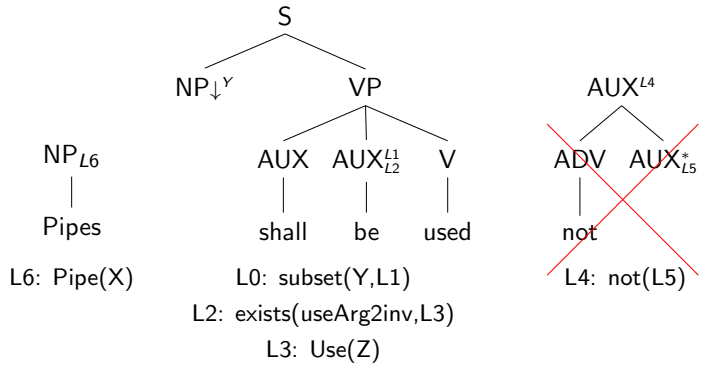
Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall not be used.	



Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall not be used.	

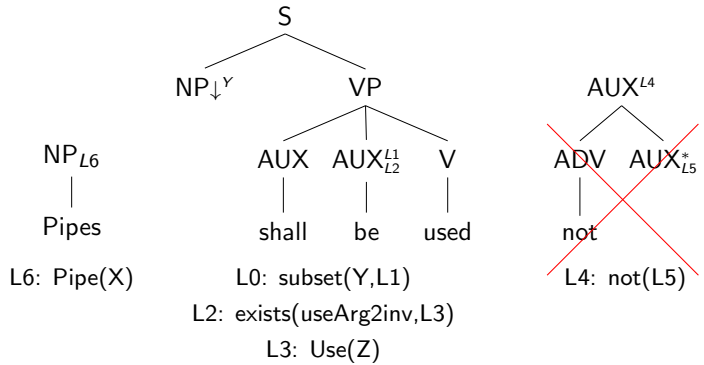


Parsing :

Pipes shall not be used. L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)

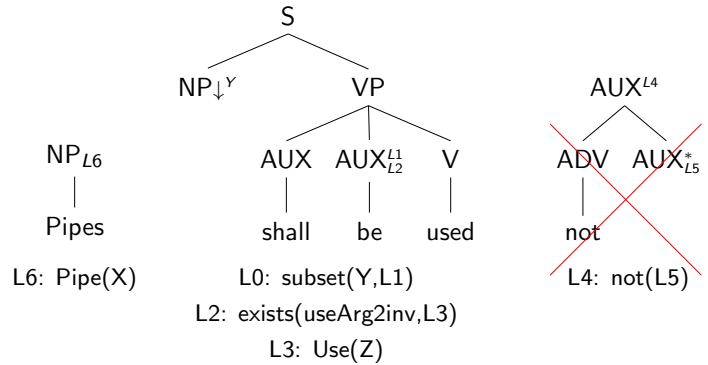
Pipes shall be used. L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)

Pipes shall ~~not~~ be used.



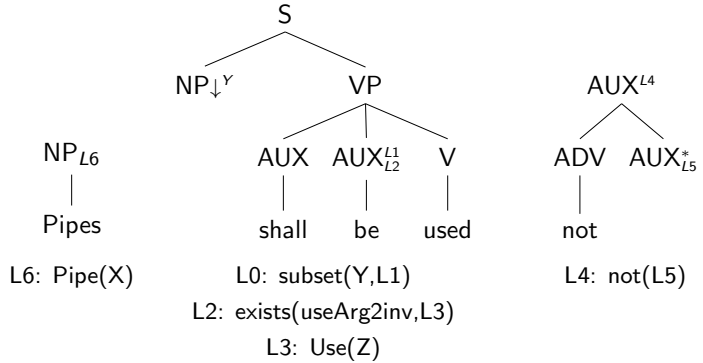
Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)



Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z) Partial Parse

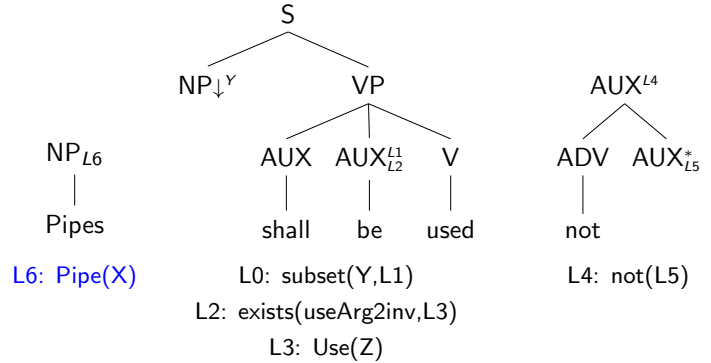


Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)

Generation :

L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)

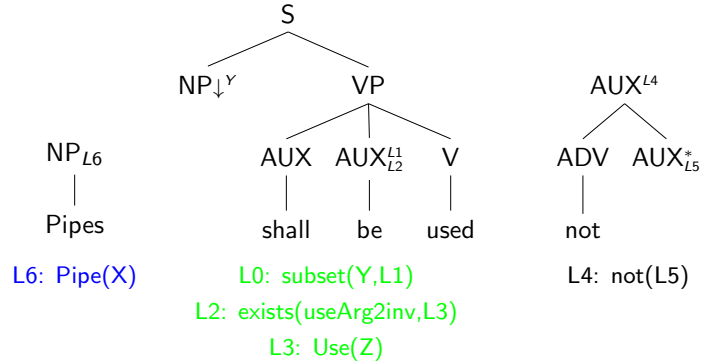


Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)

Generation :

L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)



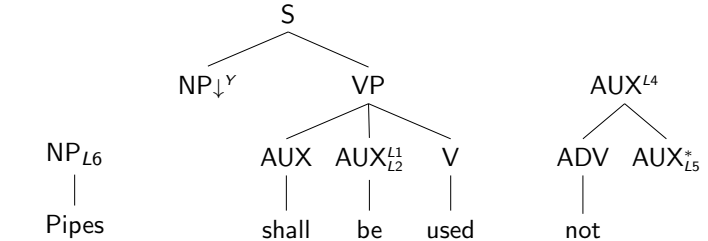
Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)

Generation :

L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)

Parsing and Generation



Parsing :

Pipes shall not be used.	L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)
Pipes shall be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)
Pipes shall ~~not~~ be used.	L6:Pipe(X) L0:subset(L6,L5) L5:exists(useArg2inv,L3) L3:Use(Z)

Generation :

L6:Pipe(X) L0:subset(L6,L1) L5:exists(useArg2inv,L3) L3:Use(Z) L1:not(L5)

Conversion Rules

- To transform a Flat Semantics formula, ϕ, to its equivalent DL formula, $\tau(\phi)$:

$$\tau(\phi) = \begin{cases} \text{ObjectSomeValuesFrom(:R } \tau(C)) & \text{if } \phi = l_i : \mathit{exists}(R, l_j) \;\; l_j : C \\ \text{SubClassOf}(\tau(C_1)\ \tau(C_2)) & \text{if } \phi = l_i : \mathit{subset}(l_j, l_k) \;\; l_j : C_1 \;\; l_k : C_2 \\ \text{ObjectIntersectionOf}(\tau(C_1)\ \tau(C_2)) & \text{if } \phi = l_i : \mathit{and}(l_j, l_k) \;\; l_j : C_1 \;\; l_k : C_2 \\ (\tau(C1) \sqcap \tau(C2)) & \text{if } \phi = l_i : \mathit{and}(l_j, l_k) \;\; l_j : C1 \;\; l_k : C2 \\ (\tau(C1) \sqcup \tau(C2)) & \text{if } \phi = l_i : \mathit{or}(l_j, l_k) \;\; l_j : C1 \;\; l_k : C2 \\ \text{not}(\tau(C)) & \text{if } \phi = l_i : \mathit{not}(l_j) \;\; l_j : C \\ \text{R}^- & \text{if } \phi = \mathit{Rinv} \\ \text{C} & \text{if } \phi = l_i : C(x) \end{cases}$$

where

- l_i are labels
- C_i are arbitrarily complex DL concepts
- R are DL roles

Data

960 SIDP sentences split into 2 categories :

- 456 Simple SIDPs : Main clause only.
  Eg: *Pipes shall be identified by labels.*

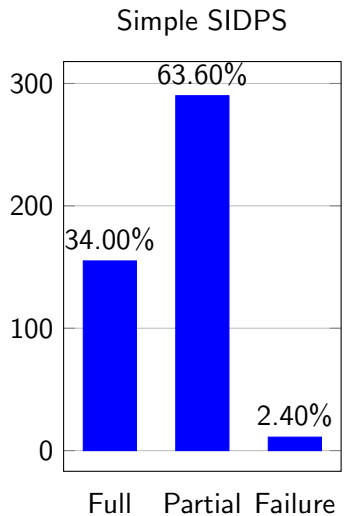
- 504 Complex SIDPs : More than one clause.
  Eg: *Object shall be qualified for continuous fuel immersion when installed inside fuel tank.*

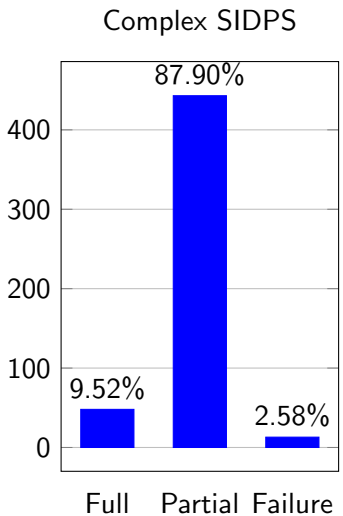
Experiment and Results

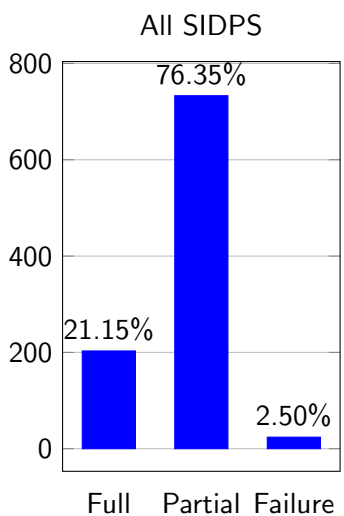
- Coverage and Robustness of the Semantic Parsing Module.
- Correctness of the derived DL formulae
  - Syntactic Correctness
  - Semantic Correctness
- Impact on the Ontology Learning Task

Coverage

- % of Sentences in each category (Simple, Complex and All) that could be parsed.

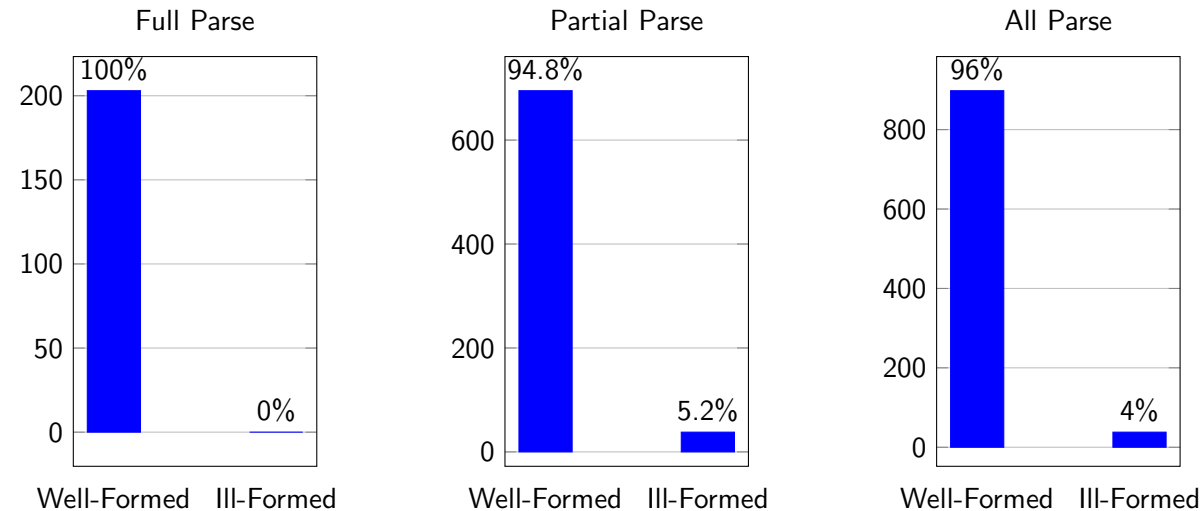






Syntactic Correctness

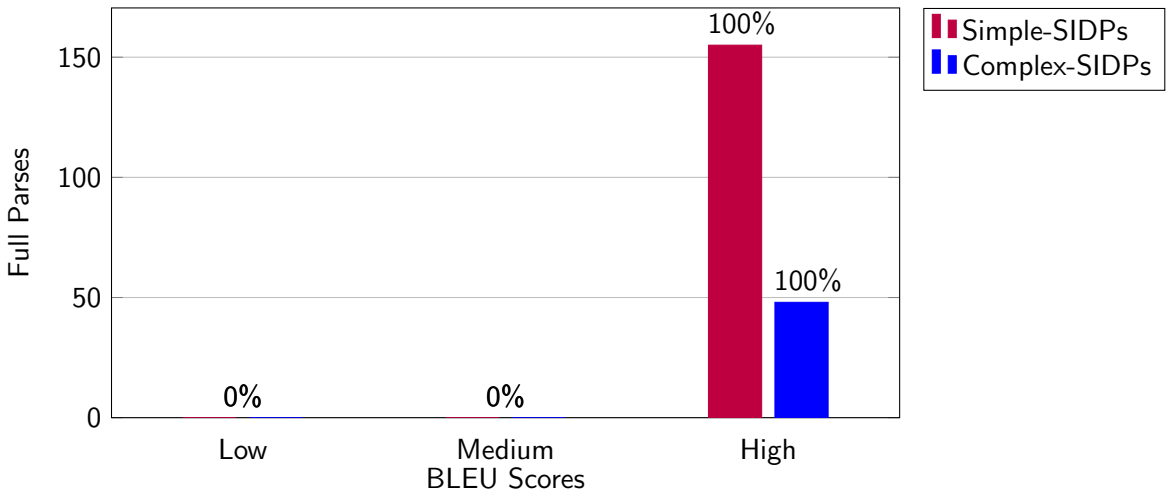
- % of Parse outputs in each category (Full, Partial and All) that are valid DL formula.



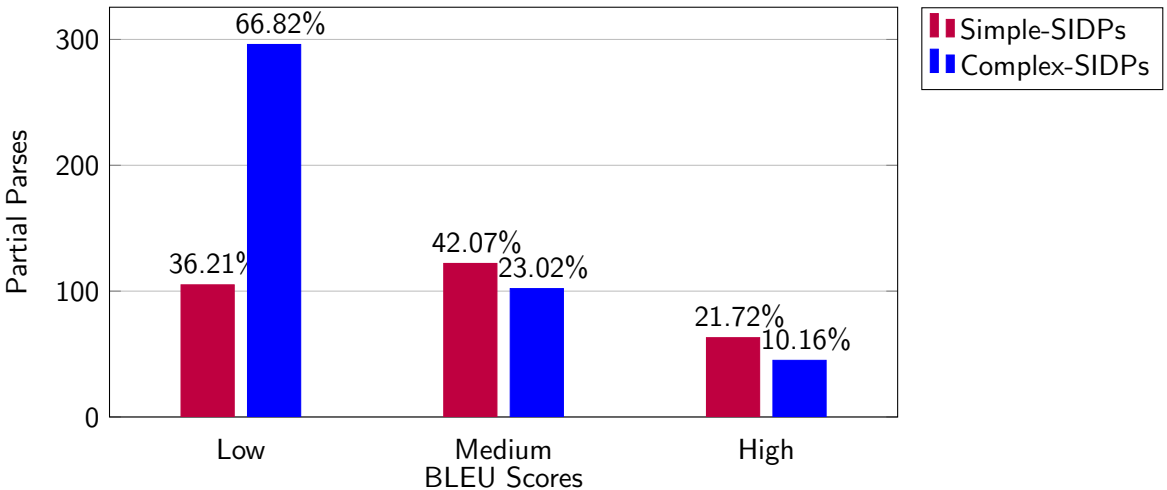
Semantic Correctness (Full Parses)

- % of Regenerated Sentence classified into BLEU categories :

Low : BLEU $\leq$ 32% Medium : 33% $\geq$ BLEU $\leq$ 66% High : BLEU $\geq$ 67%



Semantic correctness (Partial parses)



Impact on Ontology Learning

2 key steps for each axiom we derive:

- Add new Concepts and Relations found in the axiom.
- Ensure Consistency and Satisfiability before adding that axiom to the ontology.

Observations :

New classes	935
Existing classes	89
New object properties	84
Existing object properties	0
superclasses found	498
RDFS-label matches found	7
new added SIDP formulae	798 (85.3%)
rejected SIDP formulae due to syntax errors	38 (4.0%)
rejected SIDP formulae due to redundancy	91 (9.7%)
rejected SIDP formulae due to inconsistency	9 (1.0%)

Conclusions and Future Work

In Summary:

- Bridge between text and model.
- Reversability – Generation as a means of verifying Parsing.

Future Work:

- Use Reversability to build larger training corpus for Machine Learning.
- Learn the text-DL mapping using Deep Learning techniques (cf. Petrucci et al., 2016)

Thank You!

Resources (Grammar and Lexicon)

- Grammar consists of Tree Schemas rather than the trees.
- Lexical information is separately stored in a Lexicon.
- Lexicon is automatically extracted.

Semantics:
Rel = *Use*
A2 = *useArg2inv*
Tree: nx0V
Anchor: *used*
Coanchor: Aux1 → *shall/AUX*
Coanchor: Aux2 → *be/AUX*