TECHNISCHE UNIVERSITÄT DRESDEN

SEMANTIC COMPUTING

Lecture 13: Ontology Learning: Approaches

Dagmar Gromann

International Center For Computational Logic

TU Dresden, 25 January 2019

Overview

- Types and examples of ontologies
- Representation formats
- Ontology Learning in Practice

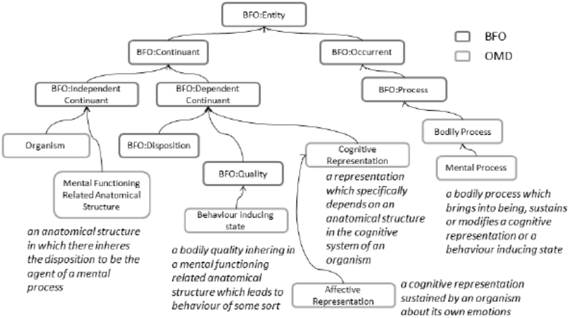
Types and examples of ontologies

Types and examples of ontologies

- Upper level ontologies
  - Basic Formal Ontology (BFO)
  - Cyc
  - Dublin core
  - Friend of a Friend (FOAF)
- Domain ontologies
  - Biomedicine: Unified Medical Language System (UMLS)
  - Finance: eXtensible Business Reporting Language (XBRL)
- Application ontologies
  - Clinical documentation: SNOMED CT
- Lexical or terminological ontologies
  - WordNet, Simple Knowledge Organization System (SKOS)
  - EuroVoc, IATE

Upper level ontologies

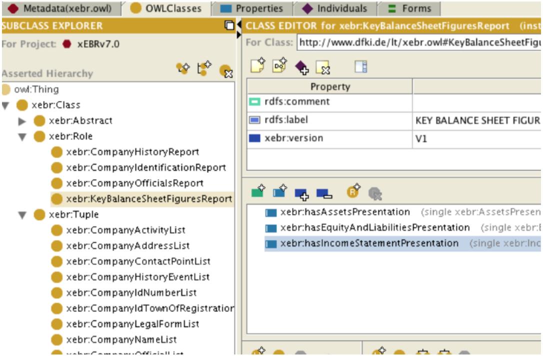
Captures concepts, relations, and axioms that apply across multiple domains, such as the Basic Formal Ontology (BFO) applied to an emotion ontology below.



Source: Hastings, J., Ceusters, W., Smith, B., and Mulligan, K. (2011). Dispositions and processes in the Emotion Ontology.

Domain Ontologies

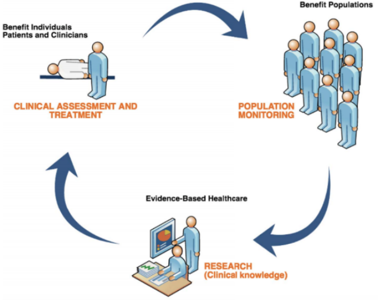
Represents concepts, relations, and axioms that are specific to a domain of discourse.



Source: Krieger, H.-U. and Declerck, T. (2013): The xEBR Ontology. Proceedings of the 26th XBRL International Conference.

Application ontologies

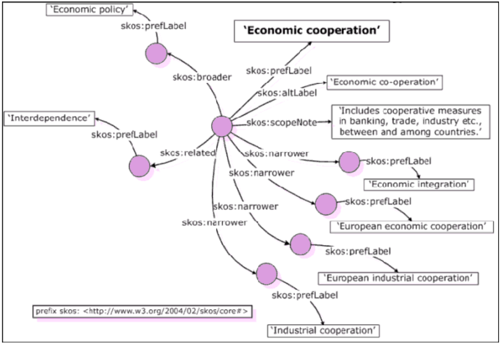
Engineered for a specific use or application scope that is specified by testable use cases, such as clinical documentation with SNOMED CT depicted below.



Source: `https://www.snomed.org/snomed-ct/five-step-briefing`

Lexical and terminological ontologies

Ontology that consists of terminological entries or synonym sets and lexico-semantic relations, as exemplified with SKOS below.

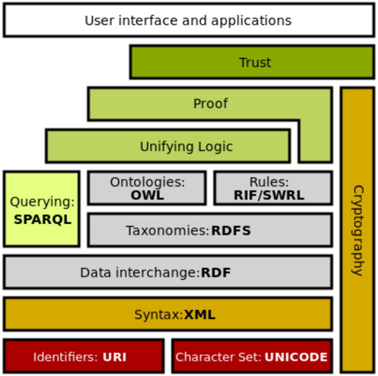


Naenudorn, E., Suphakit, N. and Chamnongsri, N. (2017). A QOS-aware Semantic Web Services Selection Model for Tourism. International Journal of Emerging Trends & Technology in Computer Science.

Representation formats

Representation formats

Semantic Web stack of technologies:



Source: `https://commons.wikimedia.org/wiki/File:Semantic_web_stack.svg`

TECHNISCHE UNIVERSITÄT DRESDEN

Resource Description Framework (RDF)

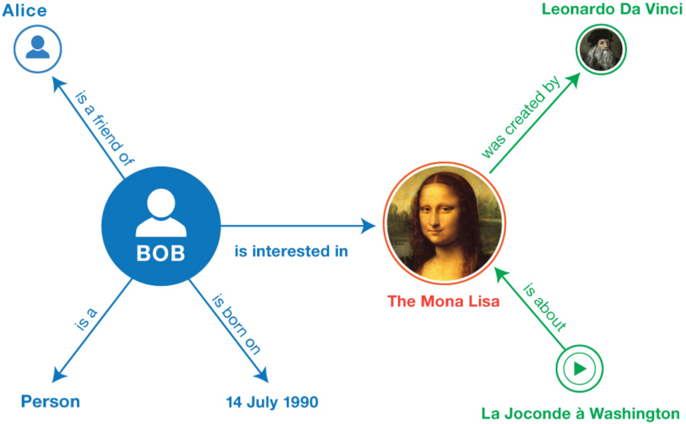
Goal: Provide a structure (framework) to describe things (resources). It consists of three basic elements that allow us to model simple ontologies:

- Resources - things being described
- Properties - relations between things
- Classes - abstract concepts used to group things

Structure = RDF triples: $<$ `Subject` $><$ `Predicate` $><$ `Object` $>$
$<$ `SemanticComputing` $><$ `hasLecturer` $><$ `DagmarGromann` $>$

All things are uniquely identified with a Uniform Resource Identifier (URI):
`http://www.foaf.com/Person#DagmarGromann`

RDF example (informal)



Source:
https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/

Web Ontology Language (OWL) Basics

- **Axioms**: basic statements that an OWL ontology expresses, e.g."it is raining", "every man is mortal" - asserted to be true)
- **Entities**: elements used to refer to real-world objects, e.g. `Class : Course`, `Student : Mary`
- **Expression**: combinations of entities to form complex descriptions from basic ones, e.g.

Entities need to be declared to be of a specific type, e.g.
`Individual : SemanticComputingObjectProperty : belongsTo`
`Class : Course`

OWL modeling

- Class Expressions:
  - Conjunction (and) : $C \sqcap D$
  - Disjunction (or): $C \sqcup D$
  - Negation (not): $\neg C$
- Property Expressions:
  - Quantifier: $\exists r.C$ (existential; some), $\forall r.C$ (universal; all)
  - Cardinality: $\geq n\ r.C$ (min), $\leq n\ r.C$ (max)
- Class Axioms:
  - Subclass: $\mathtt{Student} \sqsubseteq \mathtt{Person}$ (Student is_a Person)
  - Equivalence: $\mathtt{C} \equiv \mathtt{D}$ (C sameAs D)
  - Disjointness: $\mathtt{DisjointClasses}(\mathtt{Boy}, \mathtt{Girl})$ or $\mathtt{boy} \sqcap \mathtt{girl} \sqsubseteq \bot$
- Property Axioms: same as for class + transitive, symmetric, reflexive, functional, inverse

Web Ontology Language (OWL)

Model complex ontologies. Different types of syntaxes:

- Turtle notation: `Student rdfs : subClassOf Person`
- Manchester syntax:

  `Class : Student`

  `SubClassOf : Person`
- Description Logic (DL) syntax: $\mathtt{Student} \sqsubseteq \mathtt{Person}$
- First Order Logic (FOL) syntax: $\forall \mathtt{x}.\mathtt{Student}(\mathtt{x}) \rightarrow \mathtt{Person}(\mathtt{x})$
- OWL/XML:

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
]>
<owlx:Ontology owlx:name="http://www.example.org/person"
  xmlns:owlx="http://www.w3.org/2003/05/owl-xml">
        <SubClassOf>
                <Class URI="&example;Student"/>
                <Class URI="&example;Person"/>
        <SubClassOf
<owlx:Ontology>
```

OWL Example: Manchester syntax

```
Prefixes; Ontology imports
Declaration( NamedIndividual( :John ) )
Declaration( NamedIndividual( :Mary ) )
Declaration( Class( :Person ) )
Declaration( Class( :Woman ) )
Declaration( Class( :Man ) )
Declaration( ObjectProperty( :hasWife ) )
Declaration( ObjectProperty( :hasSpouse ) )
Declaration( DataProperty( :hasAge ) )

ObjectPropertyDomain( :hasWife :Man )
ObjectPropertyRange( :hasWife :Woman )

SubClassOf( :Woman :Person )
SubClassOf(:Man  :Person)
EquivalentClasses( :Person :Human )
DisjointClasses( :Woman :Man )

SubObjectPropertyOf( :hasWife :hasSpouse )

ObjectPropertyAssertion( :hasWife :John :Mary )
DataPropertyAssertion( :hasAge :John "51"^^xsd:integer )
```

Source: https://www.w3.org/2007/OWL/wiki/Primer#Appendix:_The_Complete_Sample_Ontology

OWL vs. RDF

- RDF: describes simple facts in form of subject-predicate-object triples (plus schema that let's you specify type)
- OWL: adds semantics to properties and classes and allows you to make statements about two things at a time (samAs, transitivity, etc.)

Ontology learning in practice

Ontology learning tasks

Bees are insects that produce honey. They have six legs. Bees live only in beehives - or just hives. Maya and Flip are bees. Maya, in particular, is a notable bee. Maya and Flip are friends.

Axiom	$\mathtt{Bee} \sqsubseteq \mathtt{Insect} \sqcap \exists \mathtt{produce.Honey}$
Relation	$\mathtt{produce(Bee, Honey)}$
Hierarchiy	$\mathtt{is\_a(Bee, Insect)}$
Concept	$\mathtt{Beehive}$
Synonym	$\{\mathtt{beehive}, \mathtt{hive}\}$
Term	$\mathtt{bee}, \mathtt{beehive}, \mathtt{hive}, \mathtt{honey}, \ldots$

Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

Ontology learning approaches

- **LExO**: Volker, J., Haase, P. and Hitzler P. (2008). Learning expressive ontologies. In Buitelaar, P. and Cimiano, P. (eds). Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Vol. 167, pp. 45-69, IOS Press
- **LearningDL**: Ma, Y. and Distel, F. (2013). Learning formal definitions for SNOMED CT from text. In Conference on Artificial Intelligence in Medicine in Europe, pp. 73-77. Springer
- **TExtual DEscription Identifier (TEDEI)**: Mathews, K. A., and Kumar, P. S. (2017). Extracting Ontological Knowledge from Textual Descriptions through Grammar-based Transformation. In Proceedings of the Knowledge Capture Conference, ACM.
- **Language to DL**: Gyawali, B., Shimorina, A., Gardent, C., Cruz-Lara, S., and Mahfoudh, M. (2017, May). Mapping natural language to description logic. In European Semantic Web Conference, pp. 273-288, Springer
- **With NMT**: Petrucci, G., Rospocher, M., and Ghidini, C. (2018). Expressive ontology learning as neural machine translation. Journal of Web Semantics, 52, 66-82.

Ontology learning approaches

Approaches until 2018:

- heavy use of NLP toolkits and corpora
- strong relying on hand-crafted rules and patterns
- targeting different source and target languages

Shared challenges:

- axiom learning challenging - mostly lightweight ontologies
- high cost of maintenance and evolution

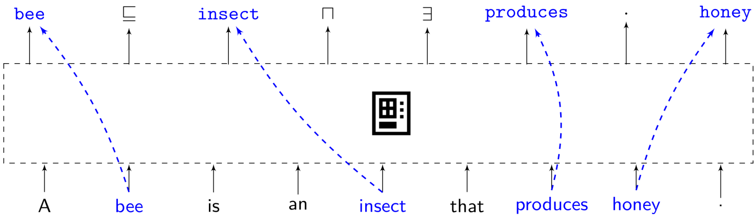
Axiom Learning: Definitions

One idea to learn formal axioms has been to benefit from intensional definitions (as opposed to extensional, object-oriented definitions):

- **Definiendum**: concept being defined (e.g. "a bee")
- **Definitor**: usually a verb introducing the definition (e.g. "is")
- **Definiens**: the genus phrase (e.g. "an insect")
- **Differentiae**: characterizations with respect to genus (e.g. "that produces honey")

Transforming NL to DL

All extralogical symbols are taken directly from the sentence:



Major challenges:

- Dataset NL to DL
- NMT architecture

Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

Dataset

We need many, good examples:

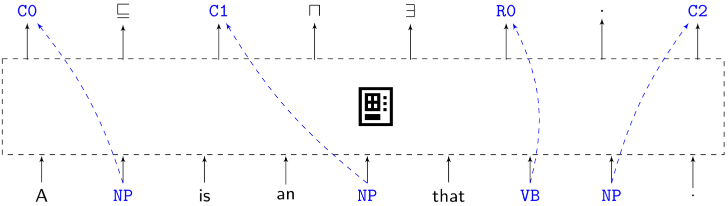
- Bees are insects that produce honey
- A bee is also an insect that produces honey
- Every bee is an insect that produces honey
- A cow is a mammal that eats grass

Dataset needs to:

- cover MANY syntactic variations of identical semantic contents
- cover many domains
- has annotated <sentence, axiom> pairs

No such dataset is currently available!

Dataset Generation



A NP is a NP that VB NP

$$C0 \sqsubseteq C1 \sqcap \exists R0.C2$$

Templates: structural regularities beyond meaning.

Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

Dataset Templates

V

Context-Free Grammar

random!

Every `C0` `R0` at least `NUM` `C1`

$\texttt{C0} \sqsubseteq \geqslant \texttt{NUM}\ \texttt{R0.C1}$

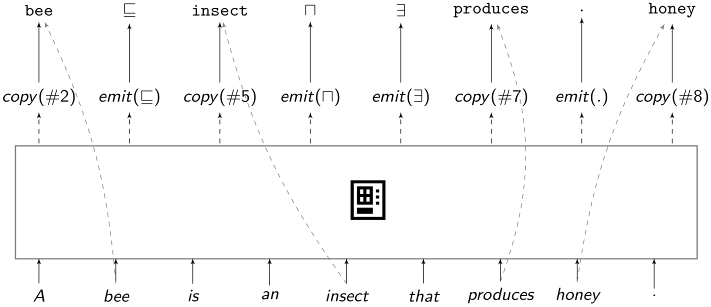
Every innocent exile craves at least 100 towers.

$\texttt{innocent exile} \sqsubseteq \geqslant 100\ \texttt{craves.towers}$

Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

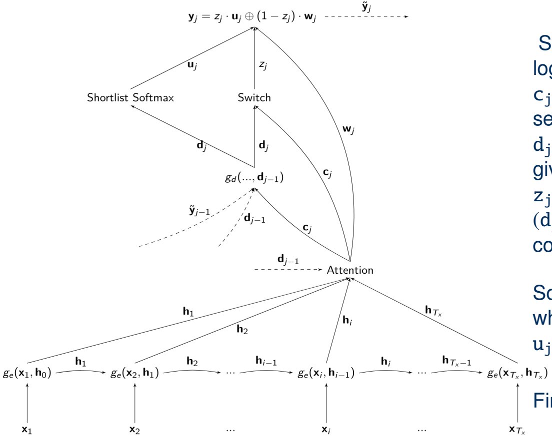
NMT processes

Train two processes: copy or emit



Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

Architecture: GRU-based



Switch network: logic or language
c_j ... summarized input sentence
d_j ... network output at a given timestep
$z_j = \sigma(W_z(d_j \oplus c_j) + b_z$
$(d_j \oplus c_j) = [d_1, \ldots, d_n, c_1, \ldots., c_n]$ concatenation

Softmax shortlist: which logical symbol
$u_j = \mathrm{softmax}(W_u d_j + b_u)$

Final output: y_j

Source: Rospocher, M. (2018). Learning Expressive Ontological Concept Descriptions via Neural Networks. SemDeep-4.

Evaluation metrics

f^k= formula with T_f symbols produced by the newtork given input sequence s^k of s^M with M number of sentences

Avg. Per-Formula Acc. $FA(\hat{\mathcal{F}}, \mathcal{F}) = \frac{CF}{M} = \frac{\sum_{k=1}^{M} \begin{cases} 1, & \text{if } f^k \equiv \hat{f}^k \\ 0, & \text{otherwise} \end{cases}}{M}$ fully automated

Avg. Edit Distance $ED(\hat{\mathcal{F}}, \mathcal{F}) = \frac{\sum_{k=1}^{M} \delta(f^k, \hat{f}^k)}{M}$ semi-automated

Avg. Per-Token Acc. $TA(\hat{\mathcal{F}}, \mathcal{F}) = \frac{\sum_{k=1}^{M} \sum_{j=1}^{T_{f^k}} \begin{cases} 1, & \text{if } f_j^k = \hat{f}_j^k \\ 0, & \text{otherwise} \end{cases}}{\sum_{k=1}^{M} T_{f^k}}$ quick control

training set size	FA	ED	TA
2000	0.61	2.48	0.92
5000	0.84	0.60	0.98
10000	**0.89**	0.47	**0.99**
20000	0.81	**0.46**	0.98

Future Directions

- Create a more varied dataset
- Test different architectures
- Combine neural architecture with knowledge representation approaches

Review of Lecture 13

- Which different types of ontologies do you know?
- Can you give a specific example for those types?
- How can ontologies be represented? What is the difference between RDF and OWL?
- What do non-neural ontology learning approaches have in common? What are main challenges?
- How does the NMT approach work? How can it be evaluated and how well did it perform?
- What is needed to perform ontology learning with deep learning?
- What could potential future directions be?