

# Semantic relations at the Machine Learning era: where have the (good old) patterns gone?

Nathalie Aussenac-Gilles  
(IRIT – CNRS, Toulouse, France)  
aussenac@irit.fr

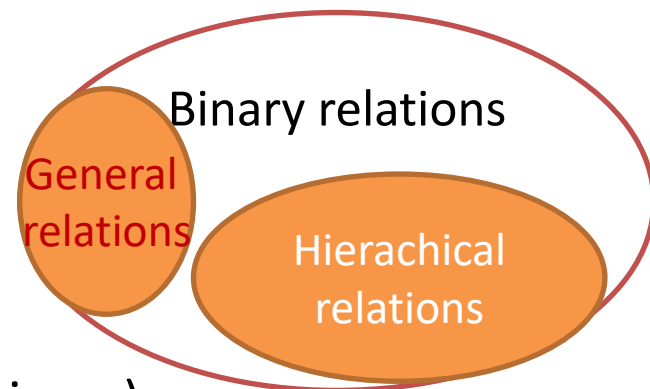


# Outline of the talk

- Semantic relations
  - What do we mean?
  - What are we looking for?
  - Why do we need them for?
- Finding semantic relations
  - Overview
  - Using patterns
  - Using machine learning
  - Towards more complementarity

# Semantic relations, what do we mean?

- **Semantic relation ... what do you have in mind?**
  - Hypernymy ... meronymy
  - Causality, temporal, spatial
  - What about
    - (Cat, *eats*, mouse)
    - (« SimplyRed », *plays*, « The right thing »)
    - (artist, *performs*, piece of music, date, location)
    - (« Eiffel Tower », *has-height*, « 324 m »)
- **Relation extraction from text: what do we have in mind?**
  - The relation is expressed in a single sentence.



# Semantic relations, what do we mean?

## Research field

- Linguistics: semantic relations, semantic roles, discourse relations
- Terminology
  - Weak structure
  - Stored in DB or SKOS models
- Information extraction

## What is a relation

A *tree* comprises at least a *trunk*, *roots* and *branches*.

A *tree* [Plants] comprises [meronymy] at least a *trunk*, *roots* and *branches*.

*tree* has\_parts *trunk*, *roots*, *branches*  
 (*tree*, has\_parts, *trunk*) ...  
 in a gardening terminology

looks for relations between instances

tree	Plantation year	Species	Branches
Tree1	1990	Oak	> 20
Tree2	1995	Oak	15

# Semantic relations, what do we mean?

## Research field

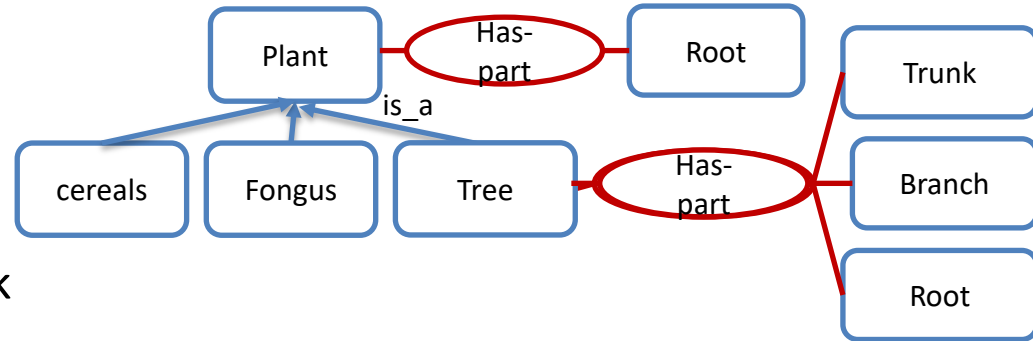
- Domain Ontology engineering
  - Formal (logic, RDF, OWL ...) and may lead to infer new knowledge
  - The relation is part of a network
  - May be shared or not
  
- Semantic web
  - Independent triples



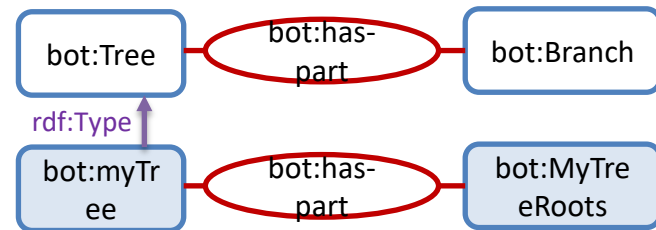
Publically available in data repositories with W3C Standard format

- Connect triples with existing ones, with web ontologies

## What is a relation



bot:Tree bot:has\_part bot:Branch

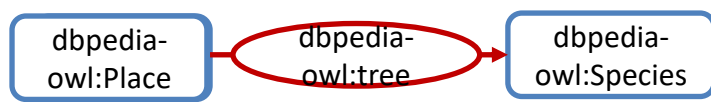


# Example: tree in DBpedia

## About: tree

An Entity of Type : Property, from Named Graph : <http://dbpedia.org/resource/classes#>, within Data Space : [dbpedia.org](http://dbpedia.org)

Property	Value
rdf:type	<ul style="list-style-type: none"> <li>▪ <a href="#">rdf:Property</a></li> <li>▪ <a href="#">owl:ObjectProperty</a></li> </ul>
<u>rdfs:domain</u>	<ul style="list-style-type: none"> <li>▪ <a href="#">dbpedia-owl:Place</a></li> </ul>
rdfs:isDefinedBy	<ul style="list-style-type: none"> <li>▪ <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a></li> </ul>
rdfs:label	<ul style="list-style-type: none"> <li>▪ tree</li> </ul>
<u>rdfs:range</u>	<ul style="list-style-type: none"> <li>▪ <a href="#">dbpedia-owl:Species</a></li> </ul>
wdrs:describedby	<ul style="list-style-type: none"> <li>▪ <a href="#">dbpedia-owl:data/definitions.xml</a></li> <li>▪ <a href="#">dbpedia-owl:data/definitions.ttl</a></li> <li>▪ <a href="#">dbpedia-owl:data/definitions.jsonld</a></li> </ul>
prov:wasDerivedFrom	<ul style="list-style-type: none"> <li>▪ <a href="http://mappings.dbpedia.org/index.php/OntologyProperty:tree">http://mappings.dbpedia.org/index.php/OntologyProperty:tree</a></li> </ul>
is <a href="http://open.vocab.org/terms/defines">http://open.vocab.org/terms/defines</a> of	<ul style="list-style-type: none"> <li>▪ <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a></li> </ul>
is <a href="http://open.vocab.org/terms/describes">http://open.vocab.org/terms/describes</a> of	<ul style="list-style-type: none"> <li>▪ <a href="#">dbpedia-owl:data/definitions.xml</a></li> <li>▪ <a href="#">dbpedia-owl:data/definitions.ttl</a></li> <li>▪ <a href="#">dbpedia-owl:data/definitions.jsonld</a></li> </ul>



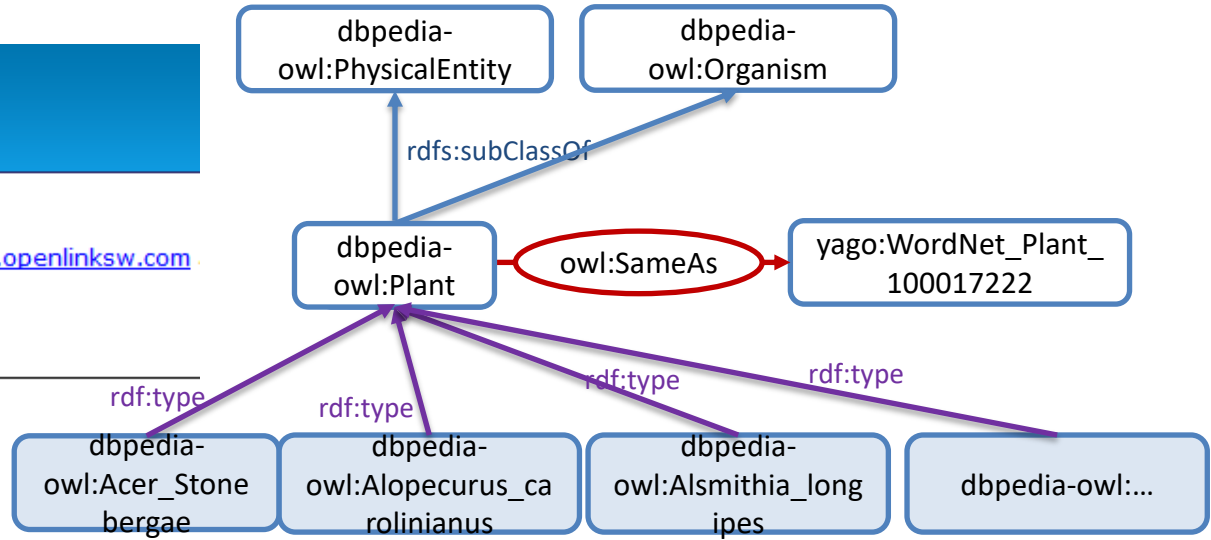
# Example: Plants in DbPedia



**About: plant** [Sponge](#) [Permalink](#)  
 An Entity of Type : [owl:Thing](#), within Data Space : [lod.openlinksw.com](#)  
 New Facets Session with This Class

Attributes	Values
<a href="#">loMismoQue</a>	<a href="#">yago:wordnet_plant_100017222</a>
<a href="#">sousClassed</a>	physical entity organism
<a href="#">label or name</a>	plant plant life flora vegetation Растение <a href="#">»more»</a>

is type of	<a href="#">Acer stonebergae</a> <a href="#">Alopecurus carolinianus</a> <a href="#">Alsmithia longipes</a> <a href="#">Amorphophallus titanum</a> <a href="#">Anderson (mango)</a> <a href="#">»more»</a>
------------	---



# Relations in the semantic web

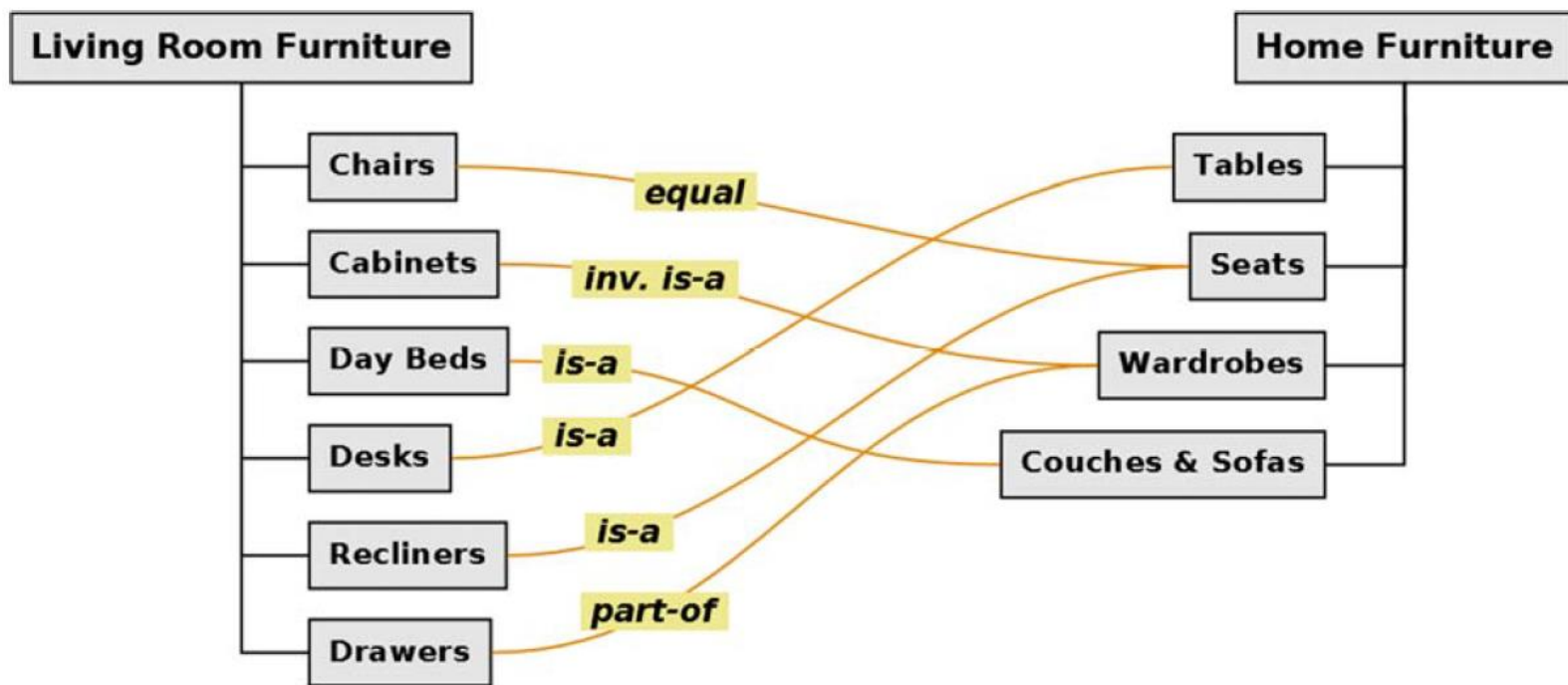
- Graph based models
  - Knowledge Vault, BabelNet, Dbpedia, Yago
  - SKOS-based representation of taxonomies, thesaurus and lexicons
- Extraction relation means
  - Identifying resources, either classes or instances
  - Giving a type to entities (hypernymy)
  - Finding and labelling relations between resources
- New types of relations: mappings
  - data integration thanks to semantic models
  - semantic representation of the content of open KB on the web
- Data linking or mapping
  - ontology alignment or
  - instance-level alignment = entity reconciliation





# STROMA Match example

Erarht Rahm <http://dbs.uni-leipzig.de/file/paris-Octob2014.pdf>



## Semantic web specificities

- Relations connect web data called resources
- Relations connect data with ontology classes: importance of hypernymy
- Relations may map ontology classes

# Semantic relations: why do we need them for?

## Research field

- **Linguistics**: to understand how language produces meaning
- **Terminology**: to capture domain specific terms and meanings
- **Natural Language Processing**: corpus annotations
- **Information extraction**: to collect structured data – the schema (classes and relations) is known

## Relations contribute to

- semantics and language interpretation
- formal semantics and discourse semantics
- **Structure terminologies and make browsing easier**
- **Connect terms**
- **Give meaning to “concepts”**
- Rich text representations for further stages in NLP workflow
- Find related entities (values) to build DB and then mine these data (statistics)

# Semantic relations: why do we need them for?

## Research field

- **Formal ontologies**: to define axioms and inferences associated with relations
- **Domain ontologies**: human and machine “understanding”
- **Linked data**: interoperability, data connection from various sites or applications



## Relations contribute to

- Inferences and reasoning
- Ex: sub-classes inherit of some class properties
- Ex : transitivity (cf. some parthood rel.), symmetry
- Ex: cardinality ...
- Relations define concepts by differences and similarities
- Relations have labels and are human readable ; each one can be processed in a specific way
- Relations connect resources (data)
- Semantics defined in ontologies (rdf:subClassOf)

# Outline of the talk

- Semantic relations
  - What do we mean?
  - What are we looking for?
  - Why do we need them for?
- Finding semantic relations
  - Overview
  - Using patterns
  - Using machine learning
  - Towards more complementarity

# Finding semantic relations

- Knowledge sources
  - human experts, text
  - existing “semantic” resources (lexicon, terminologies, ontologies, Linked Data vocabularies)
  - Domain specific vs general knowledge
- Extraction techniques from text
  - “obvious” language regularities, known relations and classes (or entities) -> **Patterns**
  - “more implicit” language regularities, medium size corpora, open list of classes/entities -> **supervised learning**
  - Very large corpora, unexpected relations -> **unsupervised learning**

# Finding semantic relations, some parameters

- Text collection(s)
  - Size, domain specific vs general
  - Structure, quality of writing
  - Textual genre (knowledge rich text?)
- Extraction techniques
  - Selection criteria
  - Complexity of implementation
  - Availability of “on the shelf” tools
- Representation
  - Input/ output format of the process
  - Nature of the semantic relation
- Validation
  - What makes a relation representation valid? Relevant?

# Historic perspective on relation extraction techniques

- Early period: around 1990
  - Patterns (Hearst, 1992) to explore definitions
  - Learning selectional preferences (Resnick)
  - Machine Learning : ASIUM (Faure, Nedellec)
  - Relations between classes
- From 2000 to 2010: more patterns, more learning
  - Association rules (Maedche & Staab, 2000)
  - Supervised Learning from positive/ negative exemples
  - Joint use of various methods (Malaisé, 2005), Text2Onto (Cimiano, Volker, 2005), RelExt
  - Relations between entities
- Since 2005: open relation extraction
  - Semi-supervised learning from small sets of data
  - Unsupervised learning: KnowItAll (Etzioni et al., 2005), TextRunner(Banko et al., 2007)
  - Distant supervision (using a KB)
  - Very large corpora (web)

# Outline of the talk

- Semantic relations
  - What do we mean?
  - What are we looking for?
  - Why do we need them for?
- Finding semantic relations
  - Overview
  - Using patterns
  - Using machine learning
  - Towards more complementarity



# A shared list of patterns: MAR-REL

- Result of the CRISTAL project
- List of candidate marks for French language and for 3 types of binary relations:
  - hypernymy, meronymy, cause
- Patterns are represented using UIMA format
- Evaluation of their precision on various corpora (domain, textual genre)

[http://redac.univ-tlse2.fr/misc/mar-rel\\_fr.html](http://redac.univ-tlse2.fr/misc/mar-rel_fr.html)

# Tuning a pattern ... an endless effort ?

```
//H25
// {appeler | nommer | qualifier de | désigner par | représenter (avec|par)
// | symboliser par | dénommer par} DET Y DET indéfini X
//

((Token{INLIST({"appeler", "nommer"}, Token.lemma)} |
 (Token.lemma == "qualifier" Token.lemma == "de") |
 (Token.lemma == "désigner" Token.lemma == "par") |
 (Token.lemma == "représenter" Token{INLIST({"avec", "par"}, Token.lemma)}) |
 (Token.lemma == "symboliser" Token.lemma == "par") |
 (Token.lemma == "dénommer" Token.lemma == "par"))
Token[0, 3]?
Token.partOfSpeech == "Det"
NominalPhrase{-> MARK(Hyponym)}
Token.lemma == "Indef"
NominalPhrase{-> MARK(Hyperonym)}{-> CREATE(Hyperonymy, "category" = "Structures d'inclusion");
```

- On appelle route nationale une route gérée par l'état.
- Sur cette carte, on symbolise par un triangle un sommet de plus de 2000m.
- Il appelle souvent son chat la nuit.
- On dénommait Louis-Philippe « la poire ».
- On appellera dans la suite de ce mémoire relation lexicale une relation qui ...

# Pattern based relation extraction, known issues

- A tree comprises *at least* a trunk, roots *and* branches.
- With branches *reaching the ground*, the willow is an ornamental tree.
- The tree of the neighbor has been delimed.
- He climbs on the branches of the tree.
- This tree is wonderful. Its branches reach the ground.
- Plant tangerine trees in a sheltered place out of the wind.
- Contains: lexical variation; **enumeration** > various parts; **modality** (exactly, at least, at most, often, ...)
- With: meronymy pattern **only in some genres** (such as catalogs, biology documents); **insertion** between the arguments
- Delimed : Term and pattern are in the same word; **implicitness**: requires background knowledge: *delimed* -> has\_part branches (and branches are cut)
- Of : Very **ambiguous** mark; polysemy reduced in [verb N1 of N2]
- Its : **reference**; necessity to take into account two sentences
- Out of: **negative form**: representation issue

# Pattern-based relation extraction, other issues

- Not enough flexibility
  - not able to handle (unexpected) variations
  - Miss find many relations
  - Need adaptation to be relevant on a new corpus
- Too strong "matching" between the sentence and the pattern itself
- Generic patterns
  - widely used with poor results (no surprise)
  - often appear as a baselines
- Building relevant domain/corpus-specific patterns is time consuming and difficult

# Outline of the talk

- Semantic relations
  - What do we mean?
  - What are we looking for?
  - Why do we need them for?
- Finding semantic relations
  - Overview
  - Using patterns
  - Using machine learning
  - Towards more complementarity

# Machine learning algorithms: Hypotheses

- A large variety of learning algorithms
  - **classification**
  - regression
  - Probabilities (naives Bayes)
  - Linear separation ...
- Classification means grouping similar objects
  - define the objects from input data
  - define a similarity measure (i.e. distance) between objects

# Learning semantic relations: hypotheses

- Prerequisite
  - Be able to isolate or (better) tag the entities to be considered as arguments
  - Text processing tools must be available (to get tokens, lemmas, POS etc.)
  - Better: identify semantic classes, compute syntactic dependencies
- Learning data
  - objects are representations built from sentences in which pairs of related « terms » occur (their context)
  - Objects are either vectors or graphs or lists of features
  - Similarity: cosine or cartesian distance or sequence alignment for graphs

# Main stages of the process

1. Preprocessing
2. Object representation
3. Training the algorithm (if supervised)
4. Running the trained model

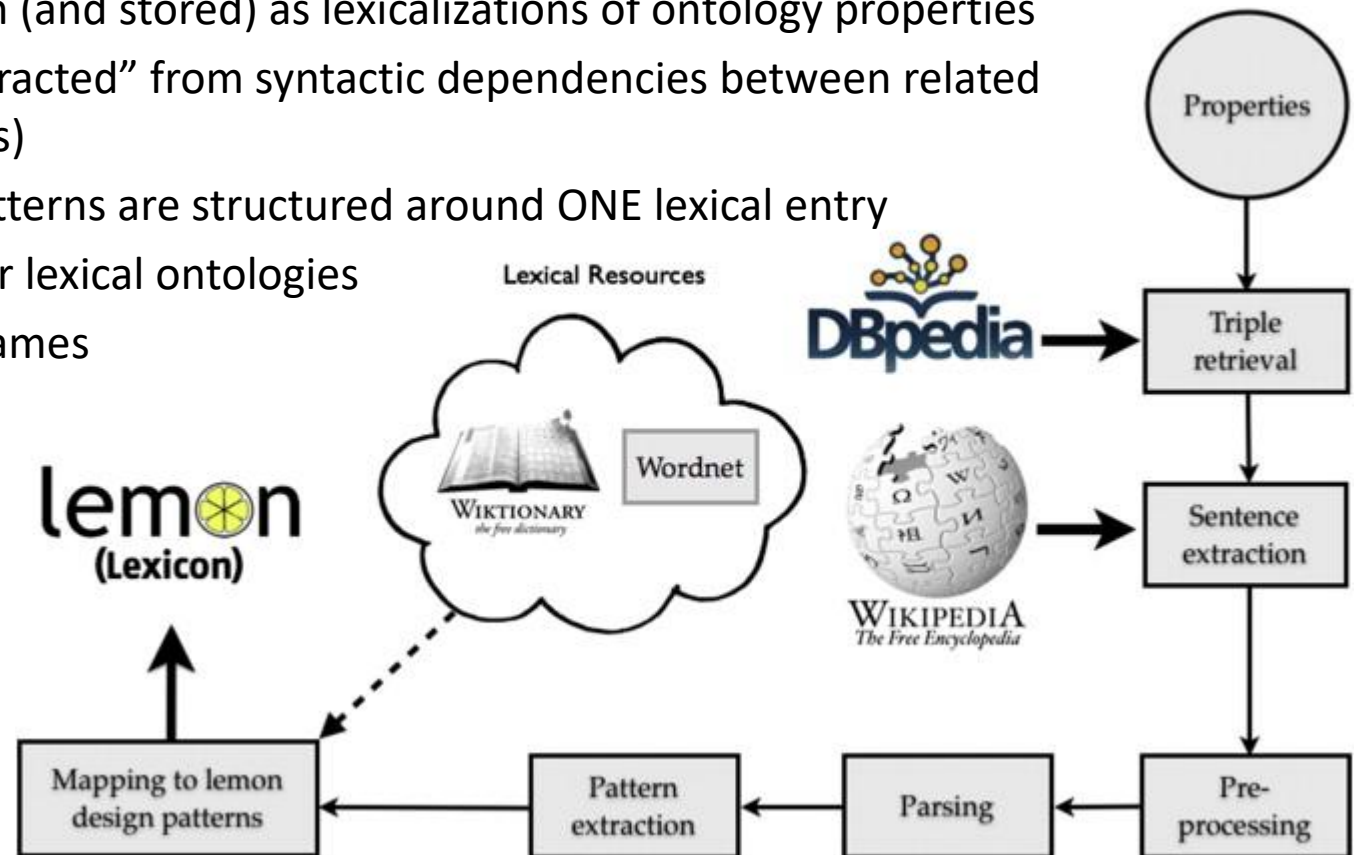


# Example1: learning patterns (1)

ATOLL—A framework for the automatic induction of ontology lexica

[S. Walter](#), [C. Unger](#), [P. Cimiano](#), DKE (94), 148-162 (2014)

- Patterns are seen (and stored) as lexicalizations of ontology properties
- Patterns are “extracted” from syntactic dependencies between related entities (in triples)
- Assumes that patterns are structured around ONE lexical entry
- Lemon format for lexical ontologies
- Entries can be frames



# Example1: learning patterns (1)

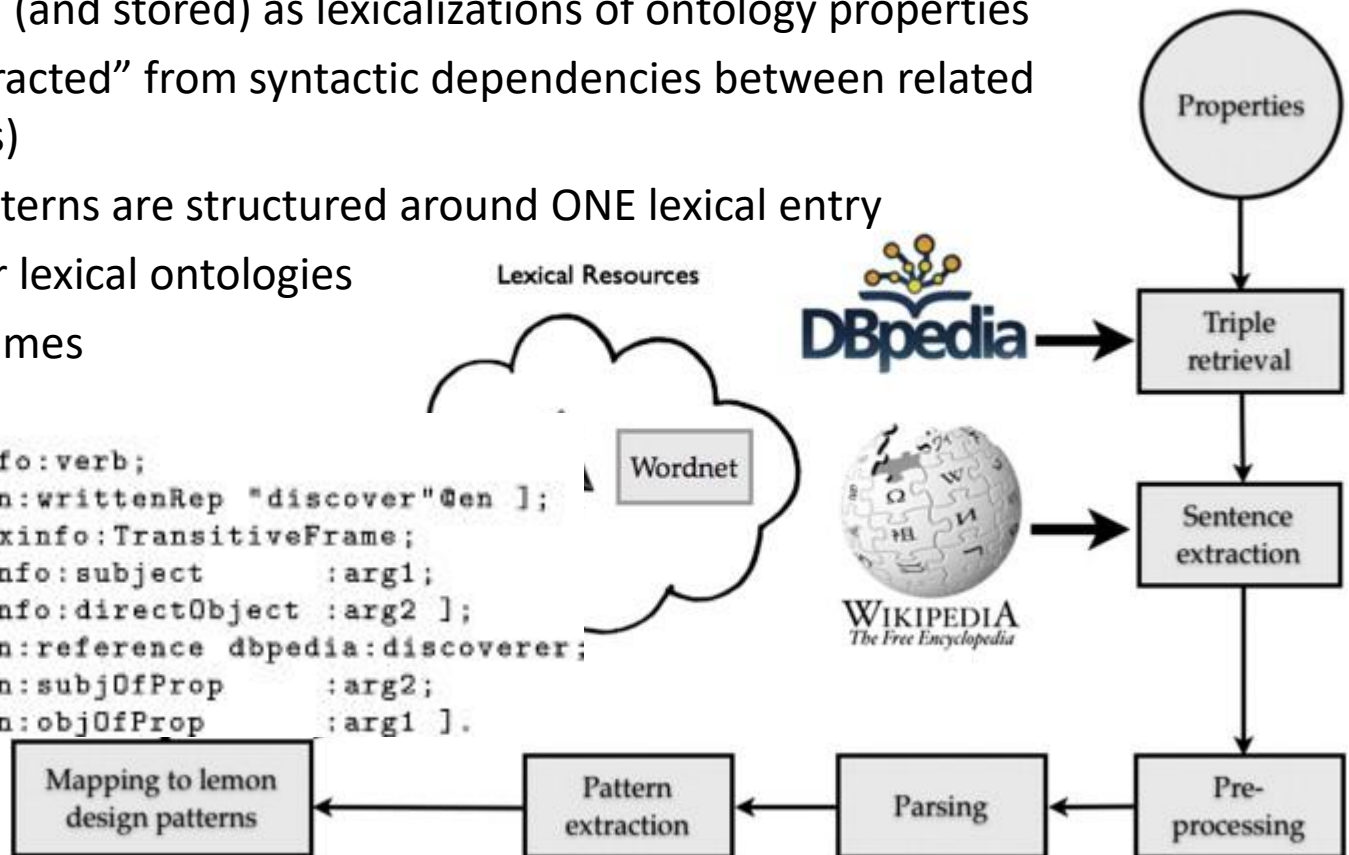
ATOLL—A framework for the automatic induction of ontology lexica

[S. Walter](#), [C. Unger](#), [P. Cimiano](#), DKE (94), 148-162 (2014)

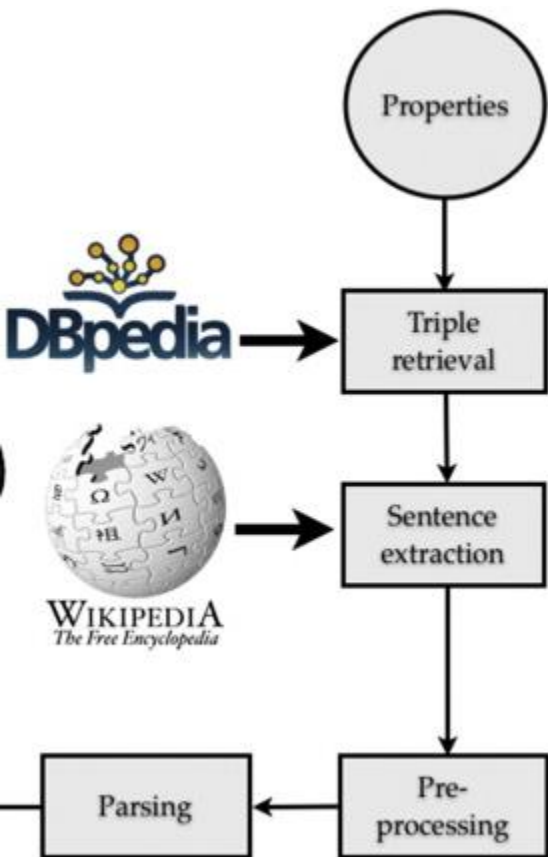
- Patterns are seen (and stored) as lexicalizations of ontology properties
- Patterns are “extracted” from syntactic dependencies between related entities (in triples)
- Assumes that patterns are structured around ONE lexical entry
- Lemon format for lexical ontologies
- Entries can be frames

```

:discover a lemon:Word;
lexinfo:partOfSpeech lexinfo:verb;
lemon:canonicalForm [ lemon:writtenRep "discover"@en ];
lemon:synBehavior [ a lexinfo:TransitiveFrame;
  lexinfo:subject :arg1;
  lexinfo:directObject :arg2 ];
lemon:sense [ lemon:reference dbpedia:discoverer;
  lemon:subjOfProp :arg2;
  lemon:objOfProp :arg1 ].
  
```



# Example1: learning patterns (2)



Dbpedia:spouse

```

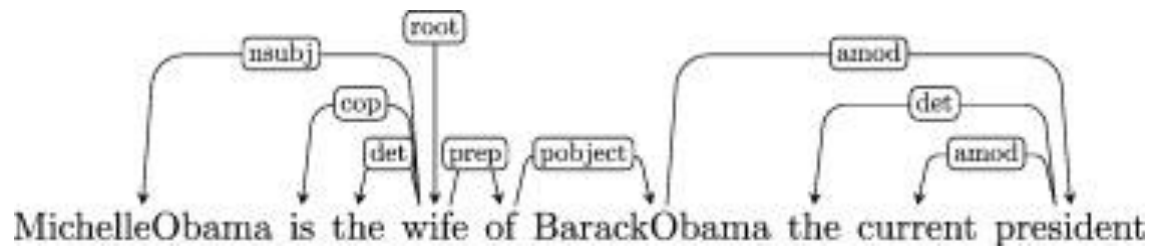
{res:Barack_Obama , dbpedia:spouse , res:Michelle_Obama}
{res:Hilda_Gadea , dbpedia:spouse , res:Che_Guevara}
{res:Mel_Ferrer , dbpedia:spouse , res:Audrey_Hepburn}
  
```

Find all lexicalizations of the entities: Michelle Obama, Mrs. Obama, Michelle Robinson ...

*Michelle Obama is the wife of Barack Obama, the current president.*

```

[ (Michelle , NNP) , (Obama , NNP) ,
  (is , VBZ) , (the , DT) , (wife , NN) , (of , TO) ,
  (Barack , NNP) , (Obama , NNP) ,
  (the , DT) , (current , JJ) , (president , NN) ]
  
```

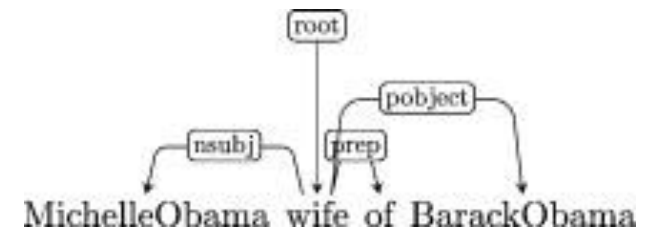


# Example1: learning patterns (3)

- Pattern = shortest path btw the 2 entities in the dependency graph

[MichelleObama (subject), wife (root), of (preposition), BarackObama (object)]

- Lexical entry in the ontology



```

:wife a lemon:LexicalEntry ;
lexinfo:partOfSpeech lexinfo:noun ;
lemon:canonicalForm [ lemon:writtenRep "wife"@en ] ;
lemon:synBehavior [ rdf:type lexinfo:NounPPFrame ;
                    lexinfo:copulativeArg :x_subj ;
                    lexinfo:prepositionalObject :y_pobj ] ;
lemon:sense [ lemon:reference dbpedia:spouse,
              lemon:subjOfProp :x_subj ;
              lemon:objOfProp :y_pobj ] .

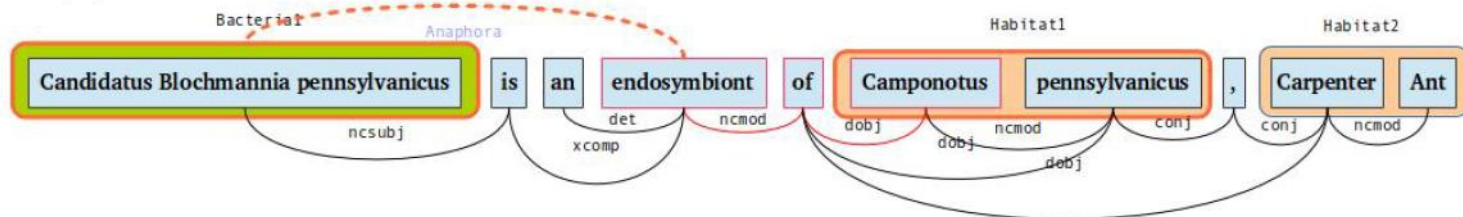
:y_pobj lemon:marker [ lemon:canonicalForm
                      [ lemon:writtenRep "of"@en ] ] .
  
```

# Example2: Learning domain specific relations using ALVIS-RE (1)

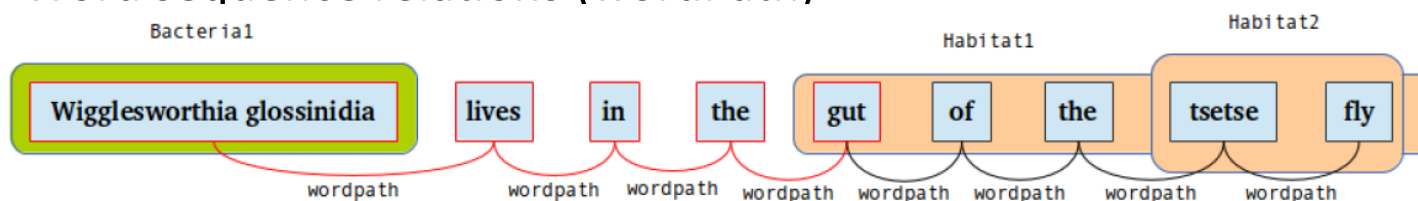
VALSAMOU D., *Automatic Information Extraction from scientific scholar to build a network of biological regulations involved in the seed development of Arabidopsis Thaliana*. ED STIC, universit  Paris Sud. Jan. 2017

## 1. Preprocessing (Tools from the AlvisNLP/ML platform)

- Tokenization in words (WoSMIG tool) and sentences (SeSMIG)
- lemmatization
- POS tagging using CCS parser
- Named Entity tagging with their canonical form



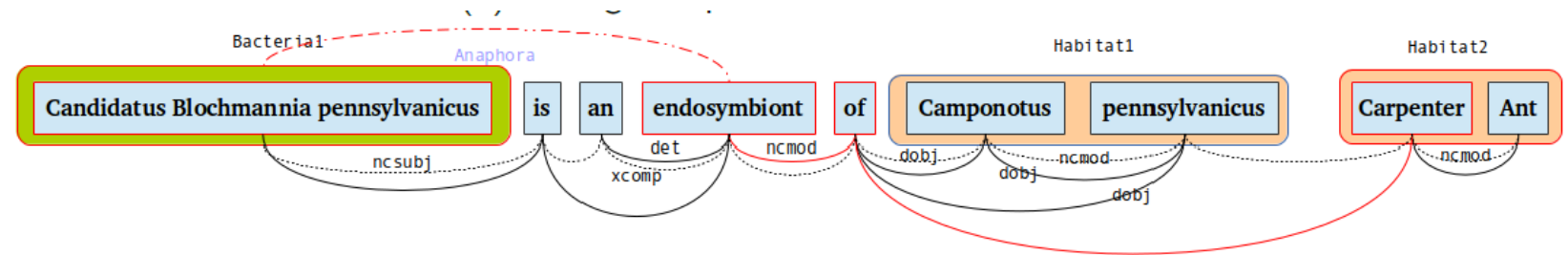
- Dependency relations (graph)
- Semantic relations are added when known (positive examples)
- Word sequence relations (wordPath)



# Example2: Learning domain specific relations using ALVIS-RE (2)

## 2. Object representation

- Representation as a path
- 3 experiments : dependencies, surface (wordPath relations) and a combination of the 2
- An algorithm finds the shortest path between the terms supposed to be Arg1 and Arg2.



(c) When finding the shortest path between Bacteria1 and Habitat2, AlvisRE prefers dependencies over wordpath relations.

# Example2: Learning domain specific relations using ALVIS-RE (3)

## 3. Object representation

- Paths are turned as sequences w,rel
- Empty nodes (gaps) are added when aligning 2 objects if needed + weight (gap penalty)
- Weights are assigned to some words

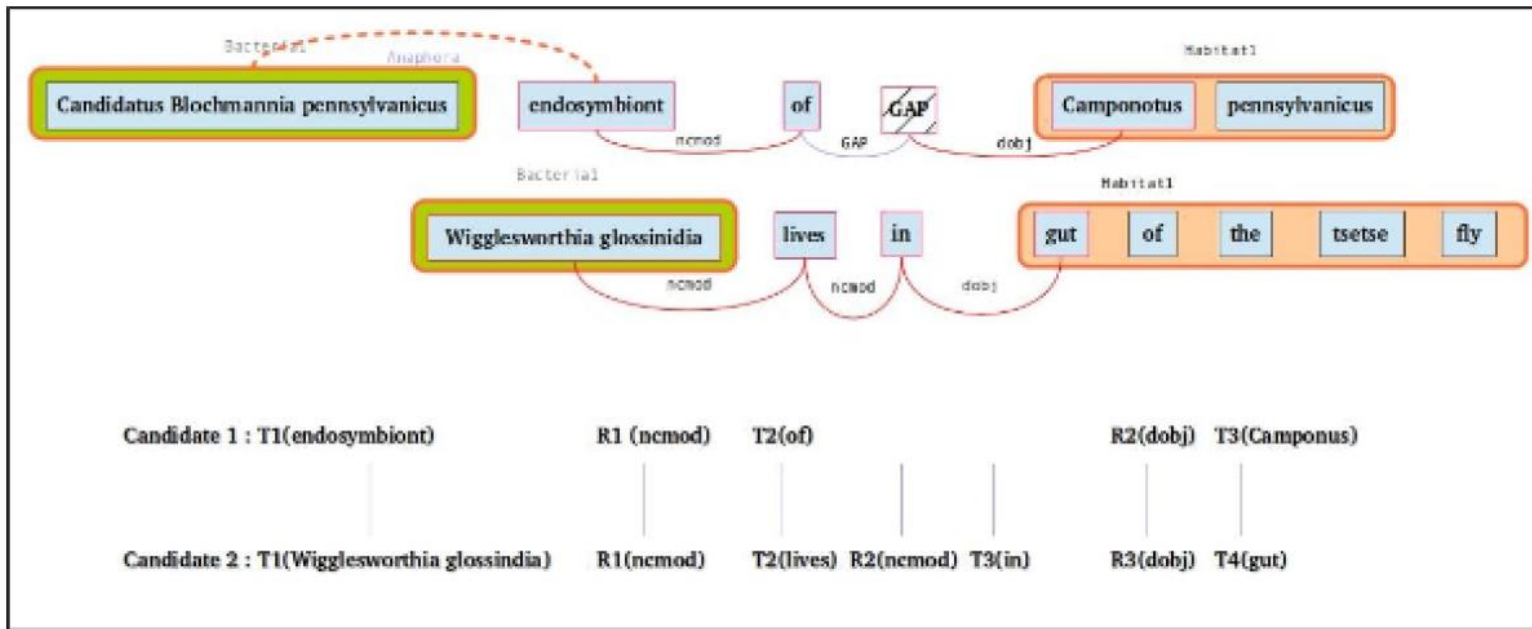


Figure 3.9: The alignment of two candidates. The second example includes an error in parsing, where *lives* has been parsed as a noun instead of a verb, and the syntactic relations linking it to its neighbours are not those of a verb object and subject. In spite of this error, AlvisRE manages to

## Example2: Learning domain specific relations using ALVIS-RE (4)

### 4. Classification

- Use SVM algorithm
- Improved using semantic information to better evaluate the similarity between the words of a sequence
  - distributional representations (DISCO or Word2Vec)
  - Classes manually related to each other
  - Classes from WordNet
- Evaluation on a real corpus



# Example3: learning relations from enumerative structures

J.-P.Fauconnier, M. Kamel. *Discovering Hypernymy Relations using Text Layout (regular paper). Joint Conference on Lexical and Computational Semantics (SEM 2015), (ACL), 2015.*

IS\_A

**Types de chaussures** [modifier | modifier le code]

**Chaussures classiques** [modifier | modifier le code]

Certaines chaussures sont portées exclusivement par les femmes :

- Escarpin, parfois nommé décolleté (il existe aussi des escarpins pour homme mais leur usage est tombé en désuétude)
- Salome
- Stiletto ou talon aiguille
- Ballerine
- Découpé

Les chaussures portées uniquement par les hommes sont le derby et le richelieu. Il existe maintenant des derbys et des richel adaptés avec un talon.

Le mocassin est porté aussi bien par les hommes et que par les femmes. Les variantes du mocassin sont le loafer (parfois non loafer. Les chaussures dont la tige recouvre la jambe sont la botte et ses variantes : la bottine, le bottillon, et la cuissarde.

**Chaussures légères et d'intérieur** [modifier | modifier le code]

Certaines chaussures ont des tiges très légères, comme :

- Sandale
- Sandalette
- Nu-pied
- Tongs
- Mule

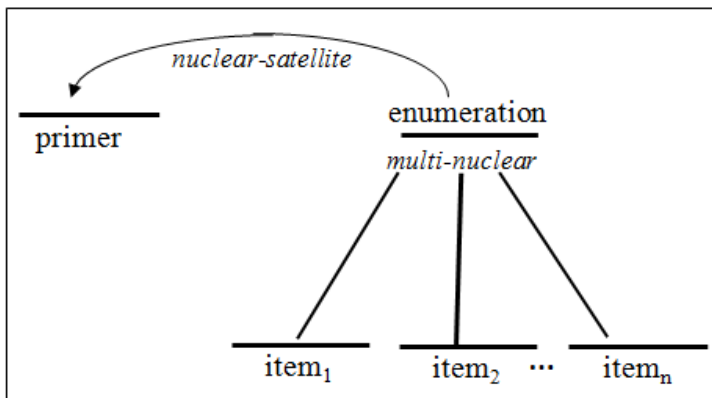
Les chaussures portées en intérieur sont :

- Pantoufle
- Charentaise
- Chausson

**Autres types de chaussures** [modifier | modifier le code]

Les rangers sont des chaussures en cuir, à long lacets, montant jusqu'aux chevilles ou plus haut. Elles sont utilisées par les armées du monde entier pour protéger les pieds contre les risques de nature électrique, chimique, mécanique, thermique (voir : chaussure de sécurité)

Les chaussures utilisées en danse sont :



Learning relations from an parallel enumerative structure =

- **classification task** to identify the relation (IS\_A, part\_Of, other)
- **Term extraction** to identify the primer and the items

# Relation extraction:

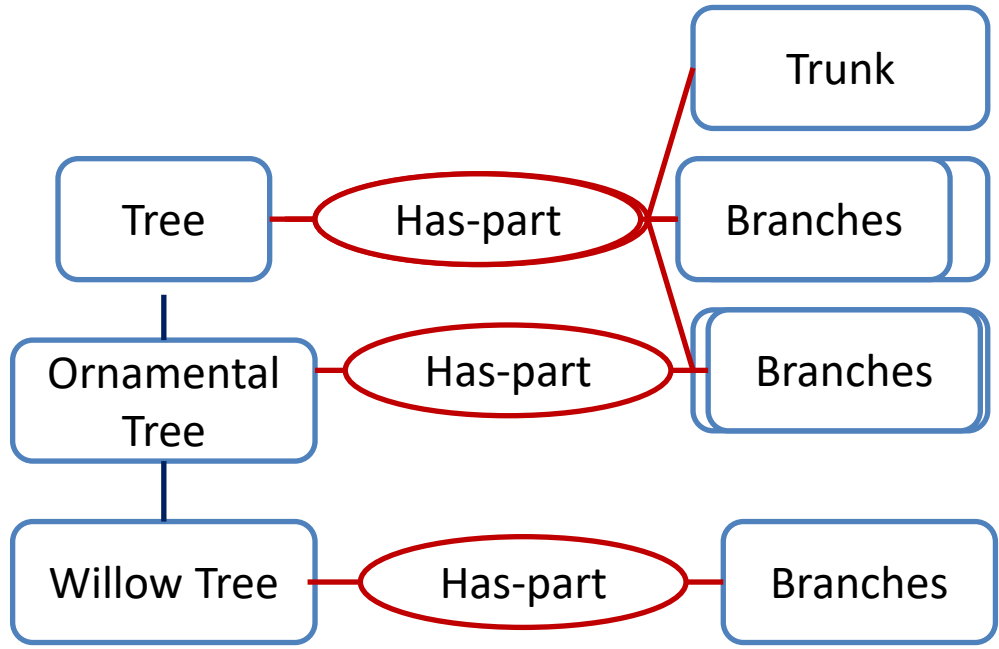
## learning relations from enumerative structures

General Features	Description
POS	The presence of a part of speech in the primer or in the items
Start/End	The first or last part of speech in the primer or in the items
Plural	Boolean indicating the presence of a plural noun
Form	The number of tokens and the number of sentences
Primer's features	
Marker	Boolean indicating the presence of a relational marker
Syntactic	Boolean indicating if the last sentence is not syntactically complete, i.e. it ends with a subordinating conjunction, a preposition, a verb, etc.
Punctuation	Returns the last punctuation

- Corpus
  - 745 enumerative structures from Wikipedia pages
  - 3 relation types: taxonomic, ontological\_non\_taxonomic, non\_ontological
- Classification task
  - Feature definition
  - Automatic evaluation of features
  - 3 algorithms are compared : SVM, MaxEntropy and baseline (majority)
  - Training of the 2 algorithms
- Results
  - 82% f-measure for SVM
  - Best result with a 2 step process (ontological yes/no -> feature and then taxonomic yes/no)

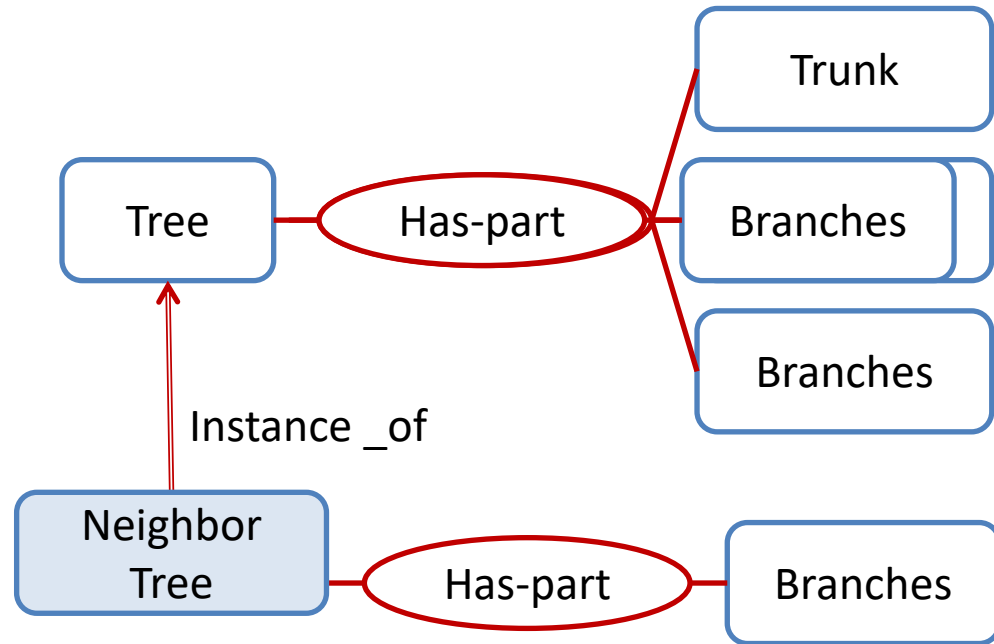
# From interpretation to representation

- A *tree* comprises at least a *trunk*, *roots* and *branches*.
- With *branches* reaching the ground, the *willow* is an *ornamental tree*.
- The *tree* of the neighbor has been delimed.



# From interpretation to representation

- A *tree* comprises at least a *trunk*, *roots* and *branches*.
- With *branches* reaching the ground, the *willow* is an *ornamental tree*.
- The *tree* of the neighbor has been delimed.
- He's climbing on the *branches* of the *tree*.
- This *tree* is wonderful. Its *branches* reach the ground.



# Finding semantic relations: what can large corpora and machine learning do for you ?

	$R_{lex}$	$P_{lex}$	$F_{lex}$	$A_{lex}$
<i>Properties with 0–1000 entity pairs</i>				
Label-based approach	0.45	0.29	0.35	0.70
Dependency-based approach	0.00	0.00	0.00	0.00
Hybrid approach	0.45	0.18	0.25	0.70
<i>Properties with 1000–10,000 entity pairs</i>				
Label-based approach	0.13	0.08	0.10	0.95
Dependency-based approach	0.36	0.08	0.13	0.81
Hybrid approach	0.50	0.06	0.11	0.77
<i>Properties with 10,000–100,000 entity pairs</i>				
Label-based approach	0.12	0.07	0.09	0.96
Dependency-based approach	0.41	0.07	0.13	0.78
Hybrid approach	0.53	0.07	0.13	0.78
<i>Properties with more than 100,000 entity pairs</i>				
Label-based approach	0.27	0.18	0.22	0.95
Dependency-based approach	0.16	0.03	0.05	0.90
Hybrid approach	0.41	0.06	0.11	0.88

- Learning patterns
  - Poor results
  - Requires very large data sets
  - Reasonable for general knowledge
- Learning relations
  - Much more relevant
  - Large variety of approaches in the state of the art
  - Key step = **select feature**
  - more features, better the results



# Outline of the talk

- Semantic relations
  - What do we mean?
  - What are we looking for?
  - Why do we need them for?
- Finding semantic relations
  - Overview
  - Using patterns
  - Using machine learning
  - Towards more complementarity

# Towards more complementarity

- the abstraction/ generalization work carried out by linguists or terminologists when designing patterns could better inspire ML-based tools
- “Patterns” can be learned automatically

Pattern can be used as

- input to define features: tag sequences matching the pattern (will become a feature) > PhD Thesis of Jean-Philippe
- an "easy method" when regularities are obvious (cf Polysemy pages in Wikipedia)
- To boot-strap and automatically identify positive examples

# Conclusion

- Context
  - complexity and diversity of what we call semantic relation extraction
  - A lot of work has been done in designing and evaluating patterns for semantic relations
- Conclusions
  - patterns cannot be the only way to look for relations
  - machine learning is now the "must-use" technique
  - machine learning requires a processing workflow



# Further readings

- Survey papers
  - N Bach, S Badaskar (2007) A review of relation extraction Language Technologies Institute, Carnegie Mellon University
  - Konstantinova T. (2014) Review of Relation Extraction Methods: What Is New Out There? in Communications in Computer and Information Science 436:15-28 · April 2014
- Recent works
  - VALSAMOU D., *Extraction automatique d'information à partir d'articles scientifiques pour la reconstruction de réseaux de régulations biologiques impliqués dans le développement de la graine chez Arabidopsis Thaliana*. ED STIC, université Paris Sud. Soutenue le 17/01/2017 Directeur de thèse : C. Nédellec. INRA, Équipe Bibliome
  - Fauconnier, J.P.. *Acquisition de liens sémantiques à partir d'éléments de mise en forme des textes : exploitation des structures énumératives*. Thèse de doctorat, Université de Toulouse, January 2016.  
[https://www.irit.fr/publis/MELODI/Fauconnier\\_These\\_2016.pdf](https://www.irit.fr/publis/MELODI/Fauconnier_These_2016.pdf)