Lexicalization Task

Given ontology and a corpus, find lexicalization for the classes and properties.
Example:
http://dbpedia.org/ontology/spouse

possible lexicalizations:

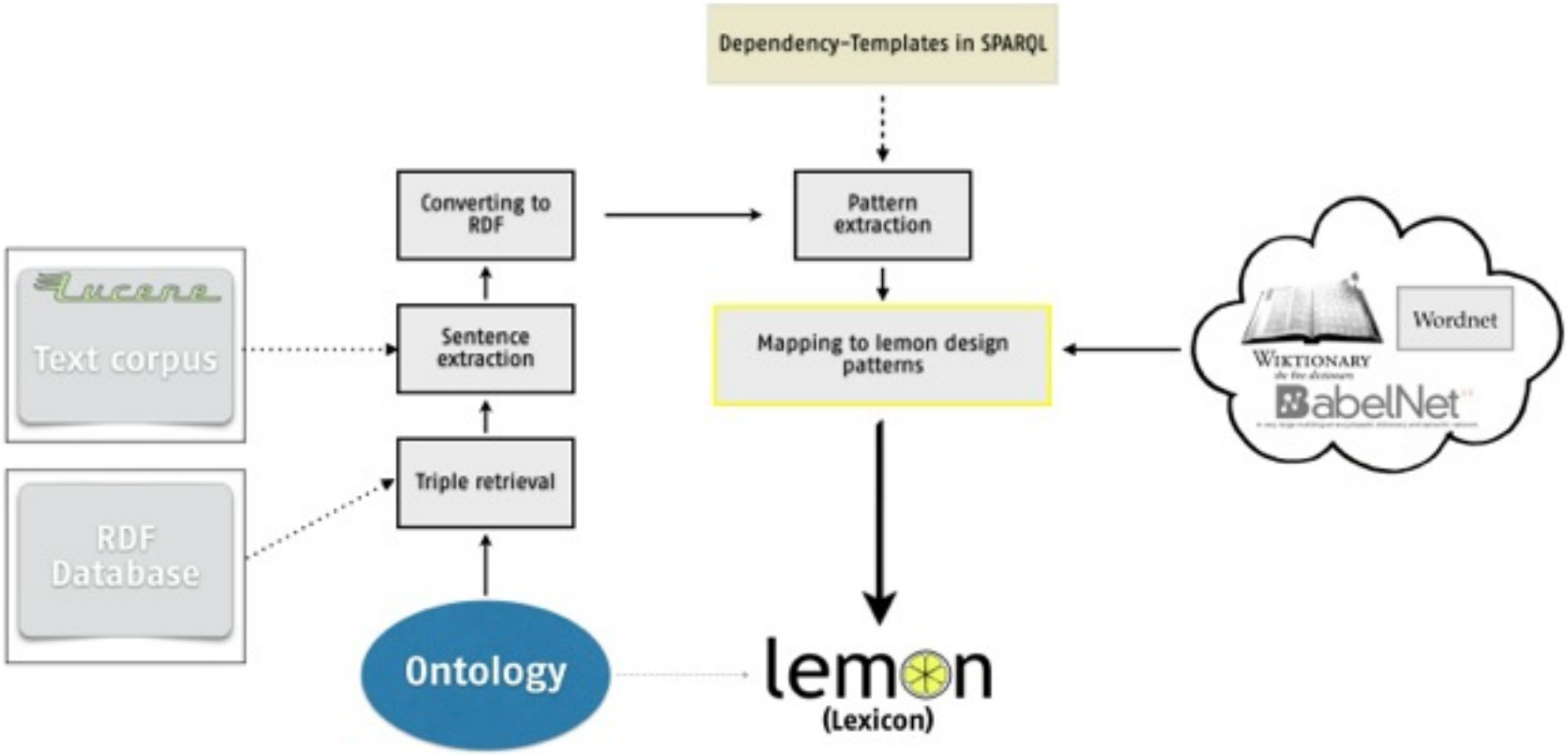
- spouse
- wife
- husband
- married to etc.

System Overview

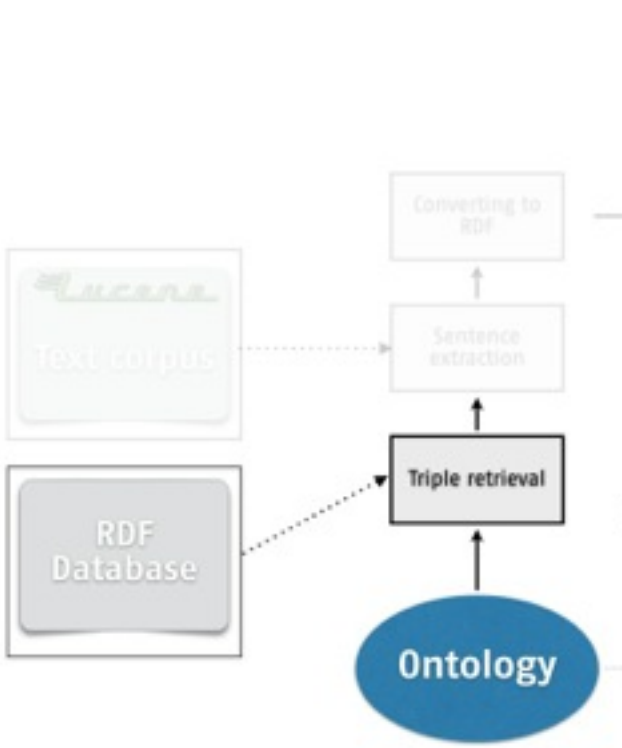
Two Approaches:

1. Label Approach
   - Taking the label of each property or class
   - Using BabelNet to find synonyms for the label
   - Creating Lemon Class entry for classes and Noun entry for properties, using label and synonyms as canonical form.
2. Dependency Approach
   - works only for properties
   - presented with more details on next slides
   - for three languages (English,German,Spanish)

Dependency Approach

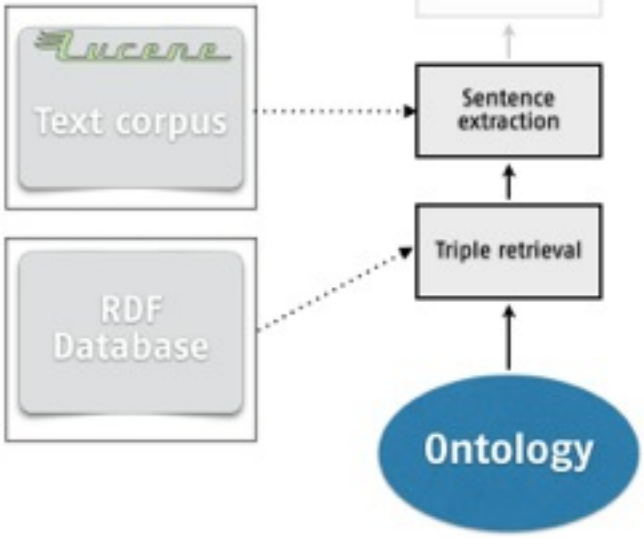


Dependency Approach – Step by Step



1. Example: dbpedia:board (875 retrieved triples)
(res:Ram_Shiram;dbpedia:board;res:Google)

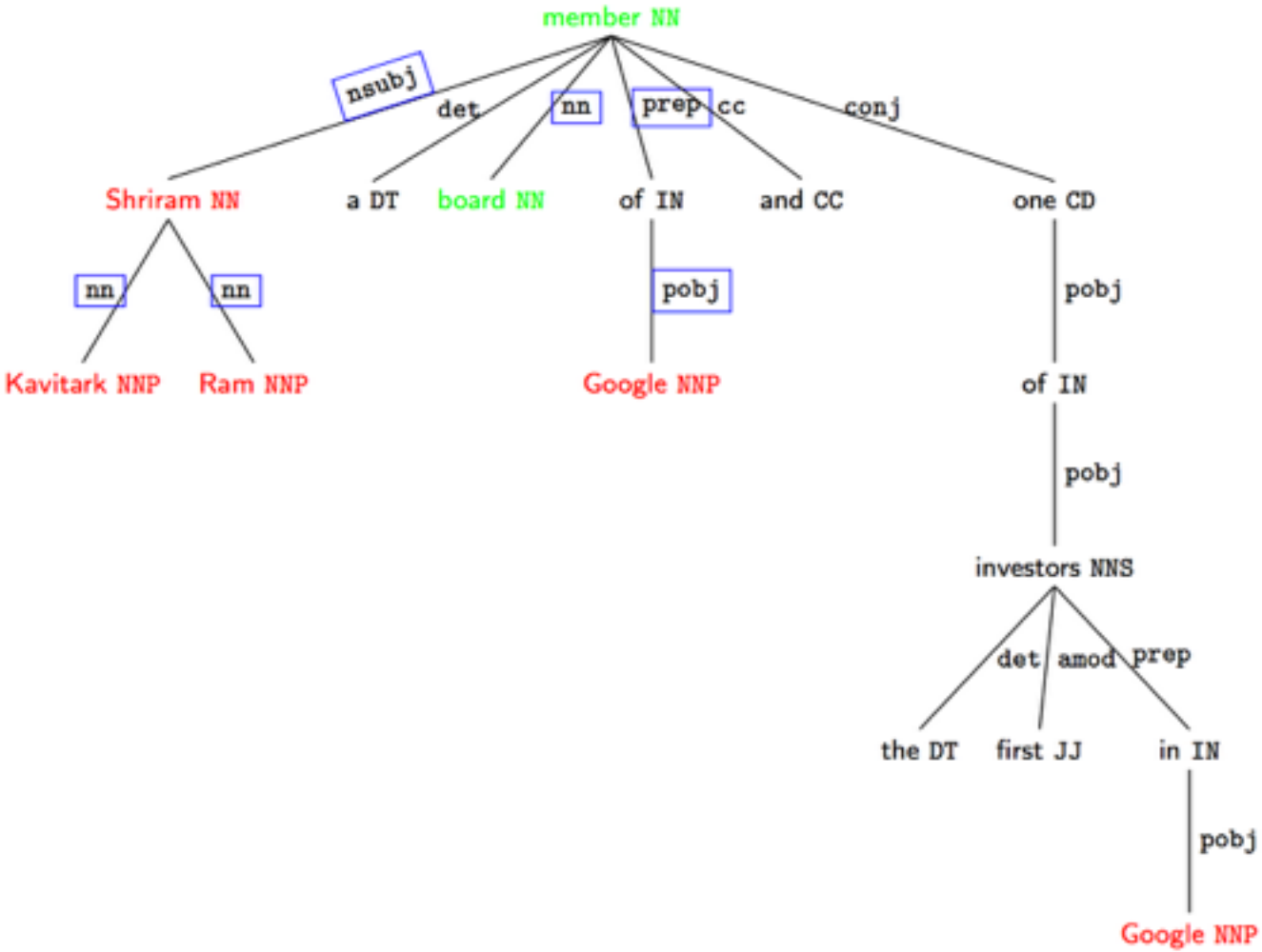
Dependency Approach – Step by Step



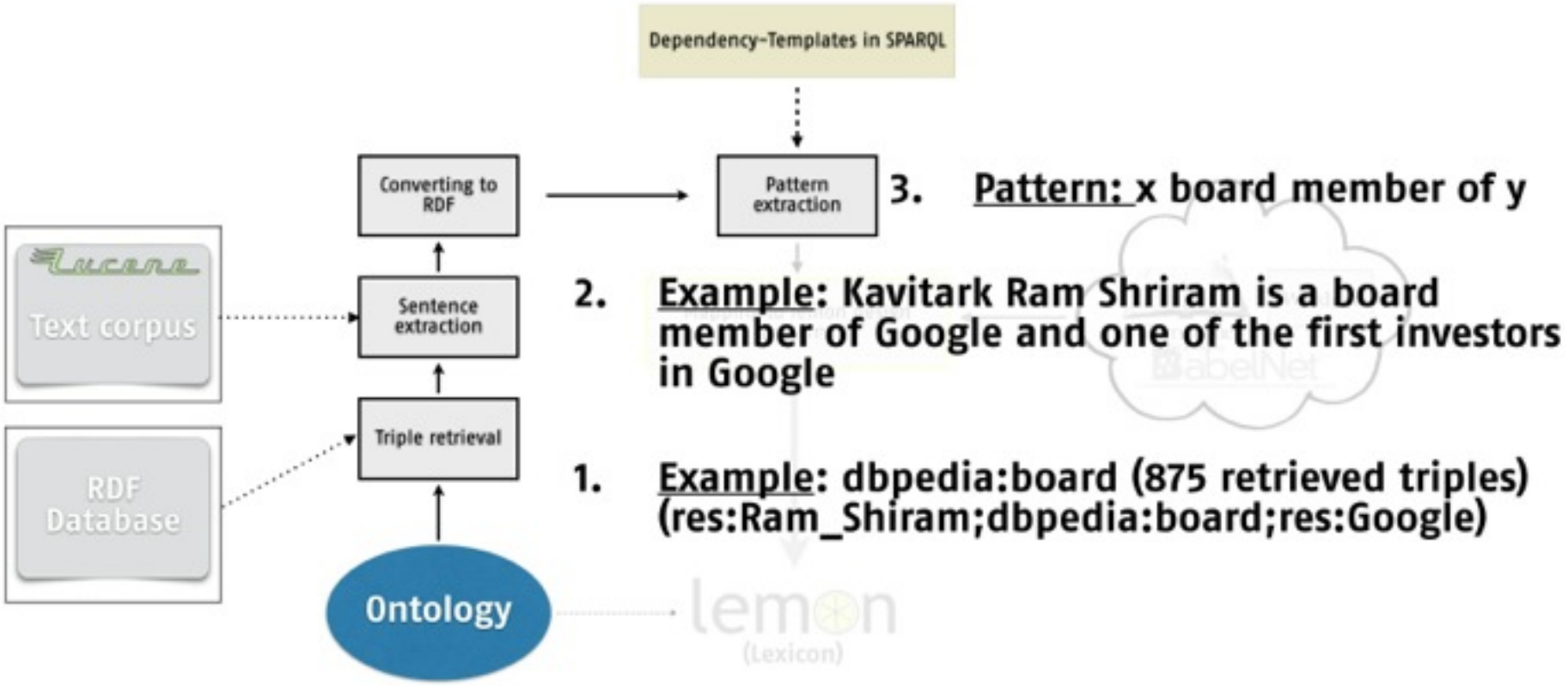
2. Example: Kavitark Ram Shriram is a board member of Google and one of the first investors in Google

1. Example: dbpedia:board (875 retrieved triples) (res:Ram_Shiram;dbpedia:board;res:Google)

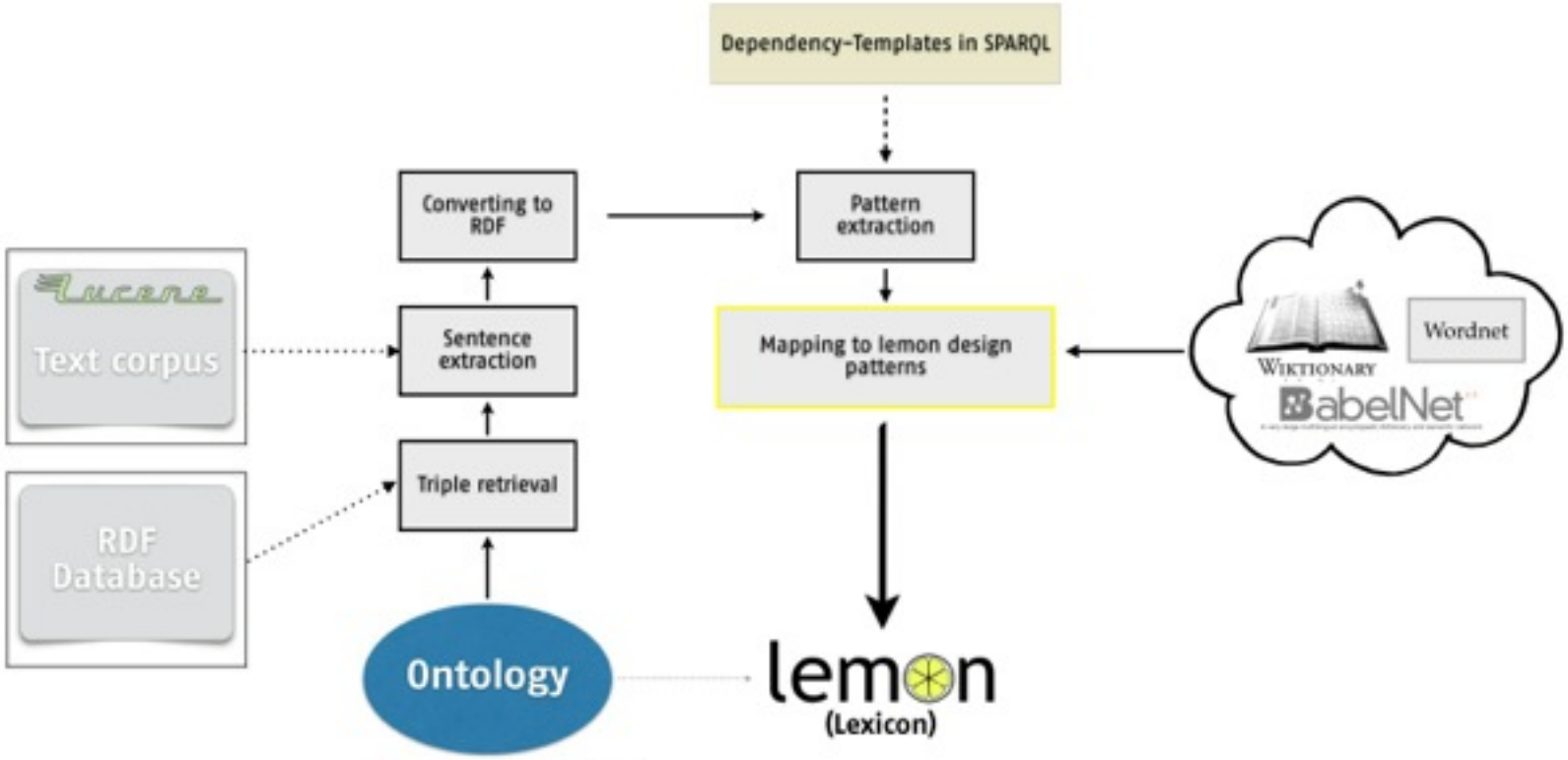
Dependency Approach – Step by Step



Dependency Approach – Step by Step



Dependency Approach – Step by Step



4. **Example:**
RelationalNoun(„board member“, dbpedia:board,
propSubj = CopulativeArg,
propObj = PrepositionalObject(„of“))

Dependency Approach – Pattern

1. Transitive verb
   - Plato **influenced** Russell.
   - Plato **beeinflusste** Russell.
   - Platón **influenció** a Russell.

2. Intransitive verb with prepositional object
   - Lincoln **died in** Washington, D.C.
   - Lincoln **starb in** Washington, D.C.
   - Lincoln **falleció en** Washington, D.C.

Dependency Approach – Pattern

3. Relational noun with prepositional object (appositive)
   - Murdoch, **creator of** the Fox Broadcasting Company, retired.
   - Murdoch, der **Gründer der** Fox Broadcasting Company, hat sich zur Ruhe gesetzt.
   - Murdoch, el **fundador** de la Fox Broadcasting Company, se jubiló.

Dependency Approach – Pattern

4. Relational noun with prepositional object (copulative construction)
   - Penelope is the **wife of** Odysseus.
   - Penelope ist die **Ehefrau von** Odysseus.
   - Penelope es la mujer de Odiseo.

Dependency Approach – Pattern

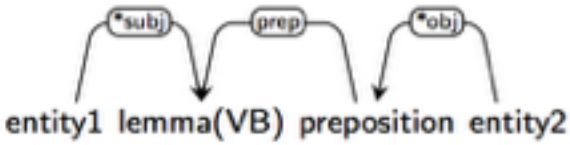
5. Relational adjective
   - Portuguese is **similar to** Spanish.
   - Portugiesisch ist **ähnlich zu** Spanish.
   - El portugués es parecido al español.

Dependency Approach – Pattern

6. Relational adjective (verb participle)
   - Audrey Hepburn **was born** in Belgium.
   - Audrey Hepburn wurde **in** Belgien **geboren**.
   - Audrey Hepburn nació en Bélgica.

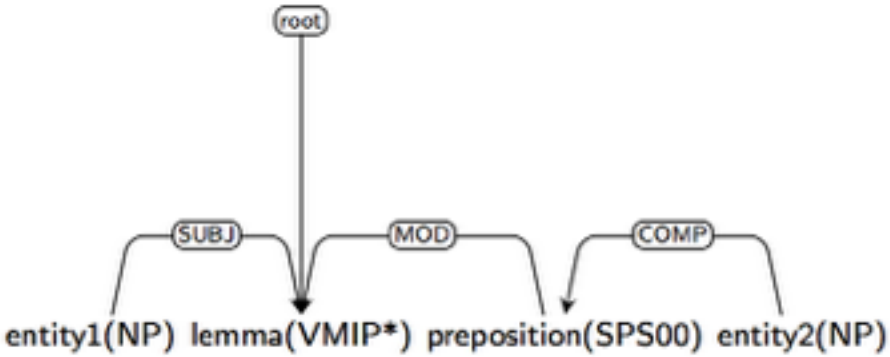
Transitive verb pattern in English

```
SELECT ?lemma ?entity1_form ?entity2_form ?prep WHERE{
?l <conll:cpostag> ?lemma_pos . "
{?l <conll:cpostag> "VB" .}
UNION"
{?l <conll:cpostag> "VBD" .}
UNION"
{?l <conll:cpostag> "VBP" .}
UNION"
{?l <conll:cpostag> "VBZ" .}
?l <conll:form> ?lemma .
?entity1 <conll:head> ?l .
?entity1 <conll:form> ?entity1_form .
?entity1 <conll:deprel> ?deprel. "
FILTER regex(?deprel, "subj") .
?p <conll:head> ?l . "
?p <conll:deprel> "prep" .
?p <conll:form> ?prep .
?entity2 <conll:head> ?p .
?entity2 <conll:form> ?entity2_form .
?entity2 <conll:deprel> ?deprel2.
FILTER regex(?deprel2, "obj") }
```



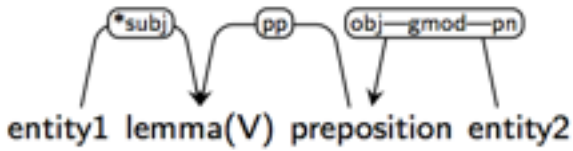
Transitive verb pattern in Spanish

```
SELECT ?lemma ?entity1_form ?entity2_form ?prep WHERE{
?l <conll:postag> ?lemma_pos . "
FILTER regex(?lemma_pos, "VMIP") .
?l <conll:deprel> "ROOT" .
?l <conll:form> ?lemma .
?entity1 <conll:head> ?l .
?entity1 <conll:form> ?e1_form .
?entity1 <conll:deprel>  "SUBJ".
?entity1 <conll:postag> "NP00000".
?p <conll:head> ?l .
?p <conll:postag> "SPS00" .
?p <conll:form> ?prep .
?p <conll:deprel> "MOD" .
?entity2 <conll:head> ?p .
?entity2 <conll:form> ?entity2_form .
?entity2 <conll:deprel> "COMP" .
?entity2 <conll:postag> "NP00000". }
```



Transitive verb pattern in German

```
SELECT ?lemma ?entity1_form ?entity2_form ?prep WHERE{
?l <conll:cpostag> ?lemma_pos .
?l <conll:cpostag> "V" .
?l <conll:lemma> ?lemma .
?entity1 <conll:head> ?l .
?entity1 <conll:form> ?entity1_form .
?entity1 <conll:deprel> "subj".
?p <conll:head> ?l .
?p <conll:deprel> "pp" .
?p <conll:form> ?prep .
?entity2 <conll:head> ?p . "
?entity2 <conll:deprel> ?e2_grammar .
?entity2 <conll:form> ?entity2_form .
FILTER( regex(?e2_grammar, "obj") || regex(?e2_grammar, "gmod") || regex(?e2_grammar, "pn")) }
```



Examples – English

Pattern	*F*
RelationalAdjective("married", dbpedia:spouse,relationalArg = PrepositionalObject("to"))	*212*
RelationalNoun("wife", dbpedia:spouse, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*149*
StateVerb("marry", dbpedia:spouse, propSubj = DirectObject, propObj = Subject)	*141*
StateVerb("marry", dbpedia:spouse, propSubj = Subject, propObj = DirectObject)	*133*
RelationalNoun("wife", dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*122*
RelationalNoun("daughter", dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*49*
RelationalNoun("daughter", dbpedia:spouse, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*38*
RelationalNoun("husband", dbpedia:spouse, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*34*
RelationalNoun("widow", dbpedia:spouse, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*22*
RelationalAdjective("born", dbpedia:spouse,relationalArg = PrepositionalObject("to"))	*22*

Examples – English

Pattern	*F*
RelationalAdjective("born",dbpedia:nationality,relationalArg = PrepositionalObject("in"))	*121*
RelationalNoun("president",dbpedia:nationality, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*36*
RelationalNoun("golfer",dbpedia:nationality, propSubj = CopulativeArg, propObj = PrepositionalObject("from"))	*26*
RelationalNoun("driver",dbpedia:nationality, propSubj = CopulativeArg, propObj = PrepositionalObject("from"))	*26*
RelationalNoun("competitor",dbpedia:nationality, propSubj = CopulativeArg, propObj = PrepositionalObject("from"))	*24*
RelationalNoun("politician",dbpedia:nationality, propSubj = CopulativeArg, propObj = PrepositionalObject("in"))	*17*
RelationalNoun("player",dbpedia:nationality, propSubj = CopulativeArg, propObj = PrepositionalObject("from"))	*13*
RelationalNoun("citizen",dbpedia:nationality, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*10*
StateVerb("return",dbpedia:nationality, propSubj = Subject, propObj = PrepositionalObject("to"))	*9*
RelationalNoun("leader",dbpedia:nationality, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*9*

Examples – English

Pattern	*F*
RelationalNoun("character",dbpedia:relative, propSubj = CopulativeArg, propObj = PrepositionalObject("in"))	*12*
RelationalNoun("ancestor",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*10*
RelationalNoun("grandfather",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*9*
RelationalNoun("son",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*8*
RelationalNoun("daughter",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*6*
RelationalNoun("grandfather",dbpedia:relative, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*6*
RelationalNoun("grandson",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*5*
RelationalNoun("sister",dbpedia:relative, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*4*
RelationalNoun("creator",dbpedia:relative, propSubj = PossessiveAdjunct("of"), propObj = CopulativeArg)	*4*
RelationalNoun("son-in-law",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct("of"))	*4*

Examples – German

Pattern	*F*
StateVerb("verheiraten",dbpedia:spouse, propSubj = Subject, propObj = PrepositionalObject("mit"))	*161*
StateVerb("verheiraten",dbpedia:spouse, propSubj = PrepositionalObject("mit"), propObj = Subject)	*155*
StateVerb("heiraten",dbpedia:spouse, propSubj = Subject, propObj = DirectObject)	*77*
StateVerb("heiraten",dbpedia:spouse, propSubj = DirectObject, propObj = Subject)	*68*
RelationalNoun("frau",dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*44*
RelationalNoun("ehefrau",dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*18*
RelationalNoun("ehefrau",dbpedia:spouse, propSubj = CopulativeArg, propObj = PrepositionalObject("von"))	*15*
RelationalNoun("gattin",dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*13*
RelationalNoun("ehefrau",dbpedia:spouse, propSubj = PrepositionalObject("von"), propObj = CopulativeArg)	*13*
RelationalNoun("gemahlin",dbpedia:spouse, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*12*

Examples – German

Pattern	*F*
RelationalNoun("nichte",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*4*
RelationalNoun("tochter",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*3*
RelationalNoun("bruder",dbpedia:relative, propSubj = PossessiveAdjunct, propObj = CopulativeArg)	*2*
RelationalNoun("schwester",dbpedia:relative, propSubj = PrepositionalObject("von"), propObj = CopulativeArg)	*2*
RelationalNoun("sohn",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
RelationalNoun("bruder",dbpedia:relative, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
RelationalNoun("cousin",dbpedia:relative, propSubj = PrepositionalObject("von"), propObj = CopulativeArg)	*2*
RelationalNoun("vater",dbpedia:relative, propSubj = PossessiveAdjunct, propObj = CopulativeArg)	*2*
RelationalNoun("schwiegervater",dbpedia:relative, propSubj = CopulativeArg, propObj = PrepositionalObject("von"))	*1*
RelationalNoun("neffe",dbpedia:relative, propSubj = PrepositionalObject("von"), propObj = CopulativeArg)	*1*

Examples – German

Pattern	*F*
StateVerb("bringen",dbpedia:source, propSubj = DirectObject, propObj = Subject)	*5*
StateVerb("münden",dbpedia:source, propSubj = Subject, propObj = DirectObject)	*4*
StateVerb("entspringen",dbpedia:source, propSubj = Subject, propObj = PrepositionalObject("am"))	*3*
RelationalNoun("quellarm",dbpedia:source, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
RelationalNoun("abfluss",dbpedia:source, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
RelationalNoun("quellast",dbpedia:source, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
RelationalNoun("führt",dbpedia:source, propSubj = CopulativeArg, propObj = PossessiveAdjunct)	*2*
StateVerb("fließen",dbpedia:source, propSubj = Subject, propObj = PrepositionalObject("vom"))	*2*
StateVerb("erzeugen",dbpedia:source, propSubj = Subject, propObj = PrepositionalObject("an"))	*2*
StateVerb("beginnen",dbpedia:source, propSubj = Subject, propObj = PrepositionalObject("am"))	*2*

Examples – Spanish

Pattern	*F*
RelationalNoun("esposa", dbpedia:spouse, propSubj = PrepositionalObject("de"), propObj = CopulativeArg)	*12*
RelationalNoun("hija", dbpedia:spouse, propSubj = CopulativeArg, propObj = PrepositionalObject("de"))	*11*
RelationalNoun("esposa", hdbpedia:spouse, propSubj = CopulativeArg, propObj = PrepositionalObject("de"))	*7*
RelationalNoun("hija", dbpedia:spouse, propSubj = PrepositionalObject("de"), propObj = CopulativeArg)	*6*
RelationalAdjective("casado", dbpedia:spouse,relationalArg = PrepositionalObject("con"))	*4*
RelationalNoun("viuda", dbpedia:spouse, propSubj = CopulativeArg, propObj = PrepositionalObject("de"))	*3*
StateVerb("conoce", dbpedia:spouse, propSubj = Subject, propObj = DirectObject)	*3*
RelationalNoun("marido", dbpedia:spouse, propSubj = PrepositionalObject("de"), propObj = CopulativeArg)	*3*
RelationalNoun("mujer", dbpedia:spouse, propSubj = CopulativeArg, propObj = PrepositionalObject("de"))	*3*
RelationalNoun("marido", dbpedia:spouse,propSubj = CopulativeArg, PrepositionalObject("de"))	*2*

Evaluation

Development: 20 classes and 60 properties (QALD-3)
Testing: 326 classes and 232 properties

	Development			Test		
	P	*R*	*F*	*P*	*R*	*F*
Dependency-based	*0.30*	*0.29*	*0.30*	*0.37*	*0.32*	*0.35*
Label-based	*0.53*	*0.24*	*0.33*	*0.56*	*0.30*	*0.40*
Both	*0.35*	*0.44*	*0.39*	*0.43*	*0.43*	*0.43*

Conclusion

- first approach for the automatic lexicalization of general ontologies in multiple languages
- Increase recall for all languages, by adding more specific pattern
- extend evaluation to complete DBpedia ontology(3.9 and 2014)
- extend Goldstandard for German and Spanish
- include and evaluate additional techniques like coreference etc.

Summary

- We introduced *lemon*
  - ✦A model for adding lexical information to ontologies
  - ✦a Concise, modular, extensible model
- *lemon* adds the following key elements
  - ✦Lexical entries
  - ✦Forms
  - ✦Senses
  - ✦Reference to ontology classes
- Ontology-lexica build on existing lexicography (dictionaries, word nets, et.c)

Summary

- Design patterns for *lemon* simplify most common usage
- Implemented as standalone language
- Can also handle
  - ✦Morphosyntactic annotations
  - ✦Argument mapping
  - ✦Multiword expressions

Summary

- Lexicalization can also be achieved automatically
- by finding existing resources
- Or, by means of dependency parsing
  - ✦ Corpus match can be found by means of SPARQL querying
- Combined approach yields F-Measure in 39%-43% range