Vincent Kríž & Barbora Hladká

RExtractor: a Robust Information Extractor

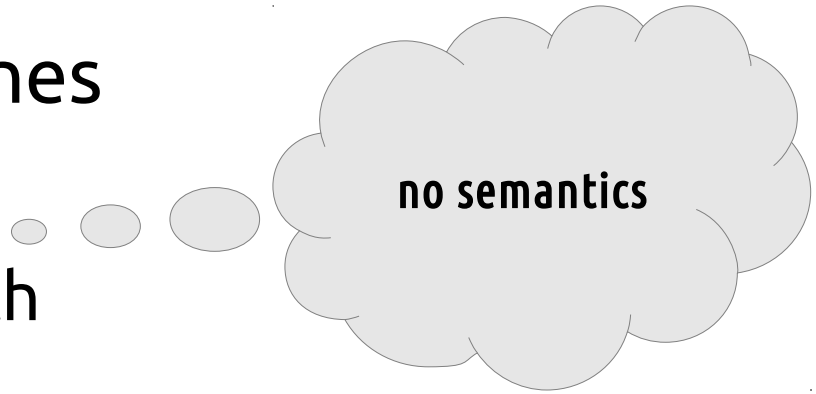
Recent developments in natural language processing
and corpus Linguistics, 11.9.2015
MFF UK

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

kriz@ufal.mff.cuni.cz
http://ufal.mff.cuni.cz/~kriz

Motivation

- large collections of documents
- efficient browsing & querying
- typical approaches
  - full-text search
  - metadata search



- semantic interpretation of documents → suitable DB & query language → user-friendly browsing & querying

Case study on legislative domain

Legal texts

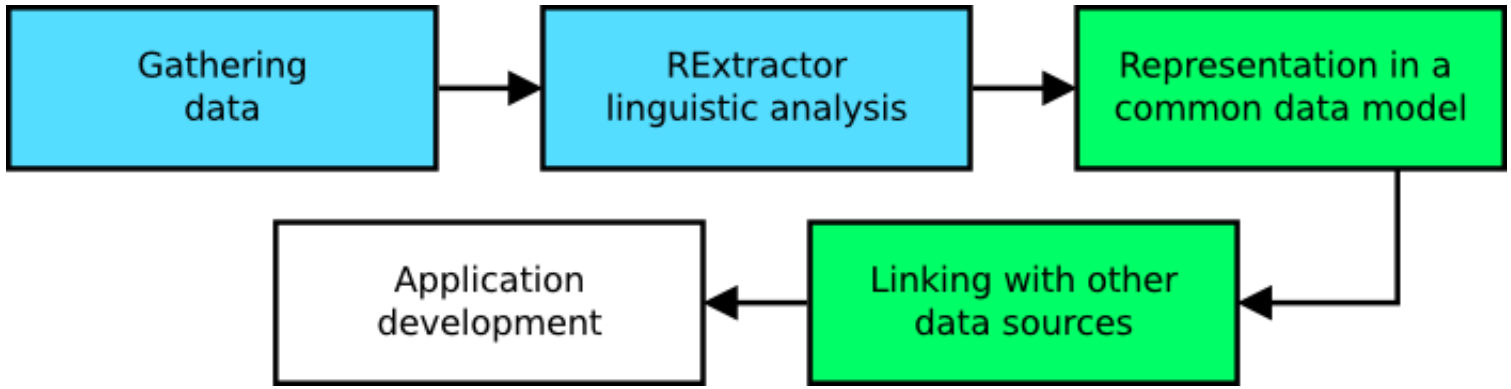
- specialized texts operating in legal settings
- they should transmit legal norms to their recipients
- they need to be clear, explicit and precise

Sentences

- simple sentences are very rare
- usually long and very complex

Legal texts are “generally considered very difficult to read and understand” (Tiersma, 2010)

Scenario



- **Cooperation between**
  - Information Extraction
  - Semantic Web

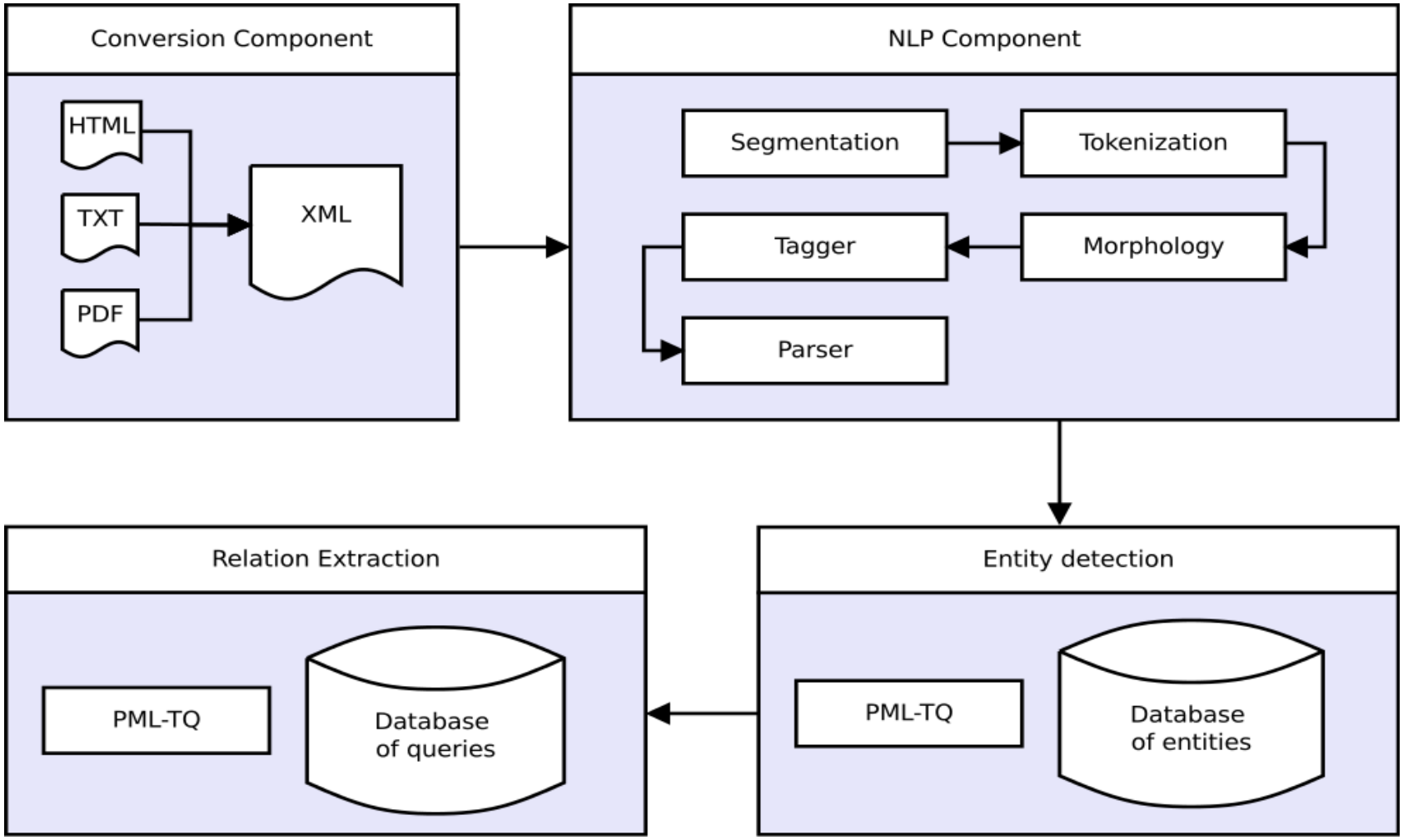
Scenario

- **Extracting knowledge base** 
  - set of entities and relations between them
  - linguistic analysis (RExtractor)
- **Knowledge base representation** 
  - Linked Data Principles
  - Resource Description Framework (RDF)

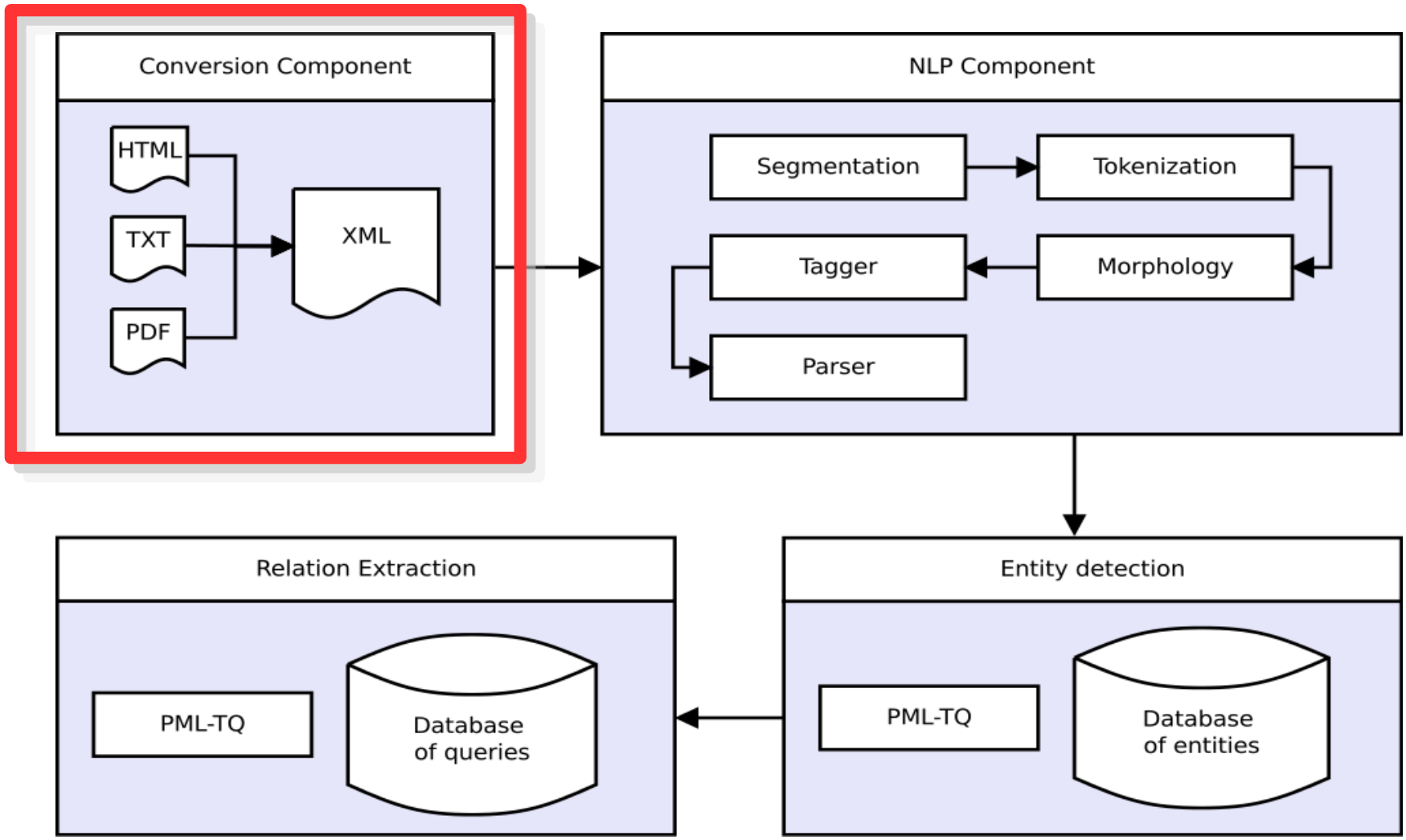
Scenario

- **Extracting knowledge base** 
  - set of entities and relations between them
  - linguistic analysis (RExtractor)
- **Knowledge base representation** 
  - Linked Data Principles
  - Resource Description Framework (RDF)

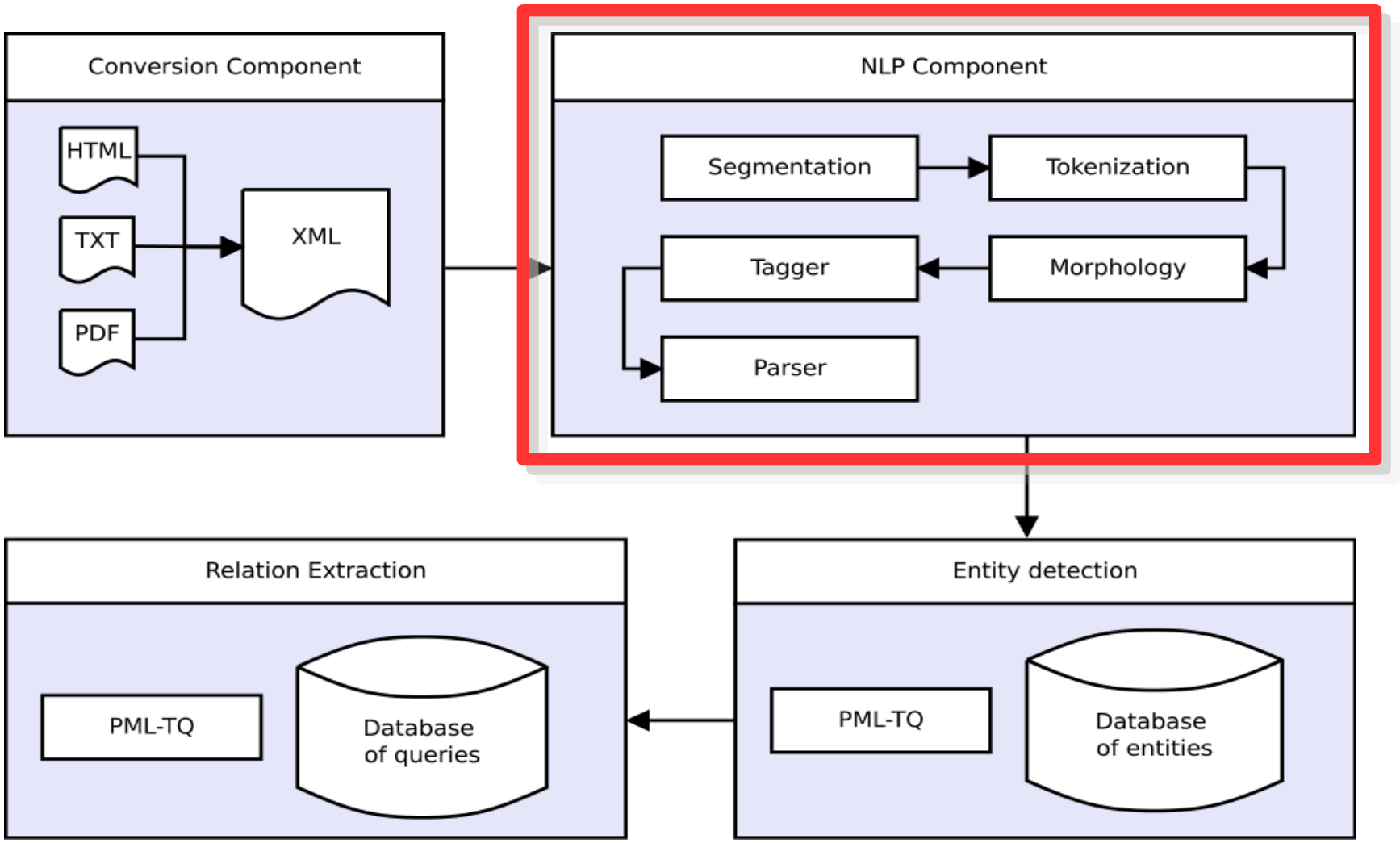
RExtractor Architecture



RExtractor Architecture



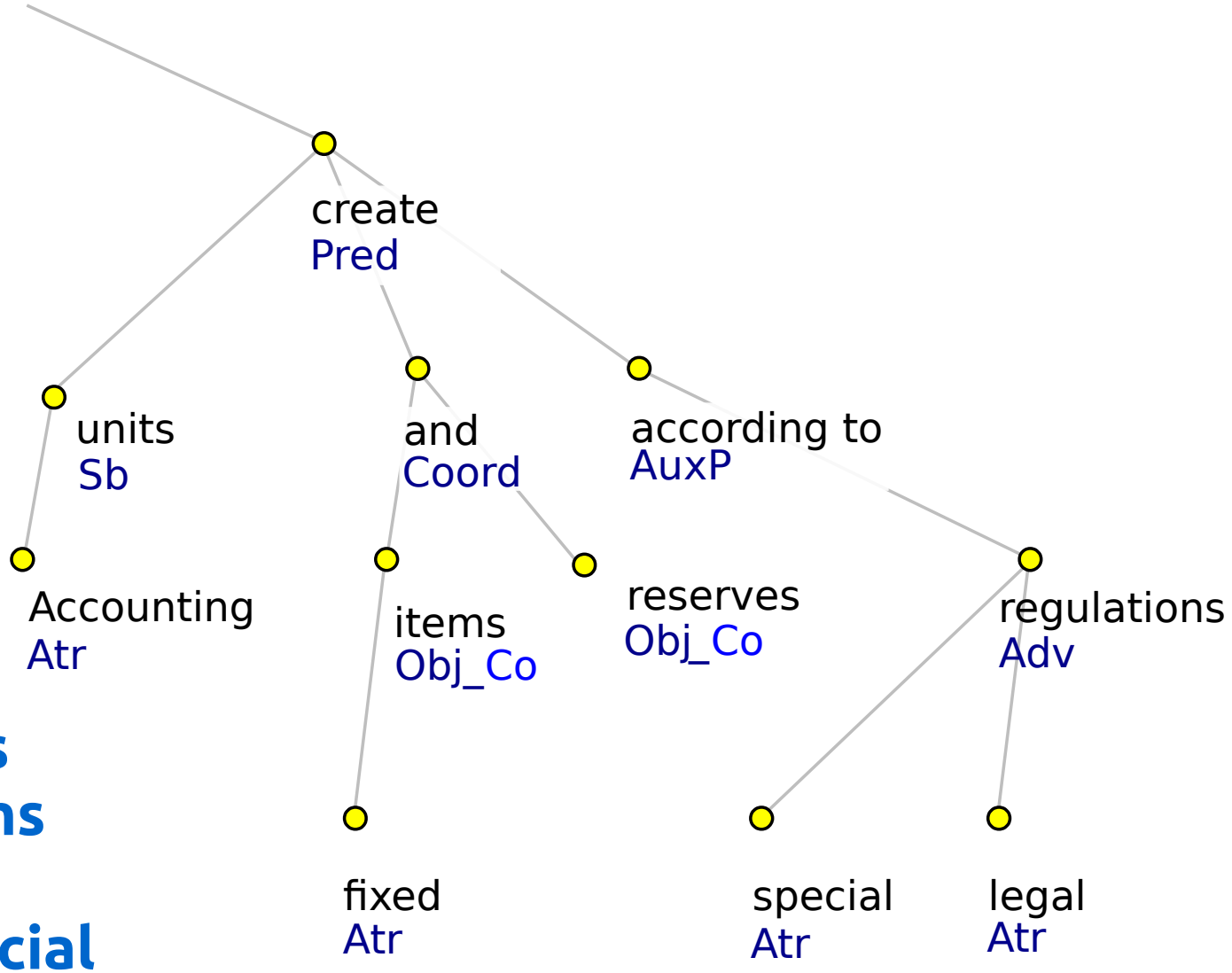
RExtractor Architecture



NLP Component

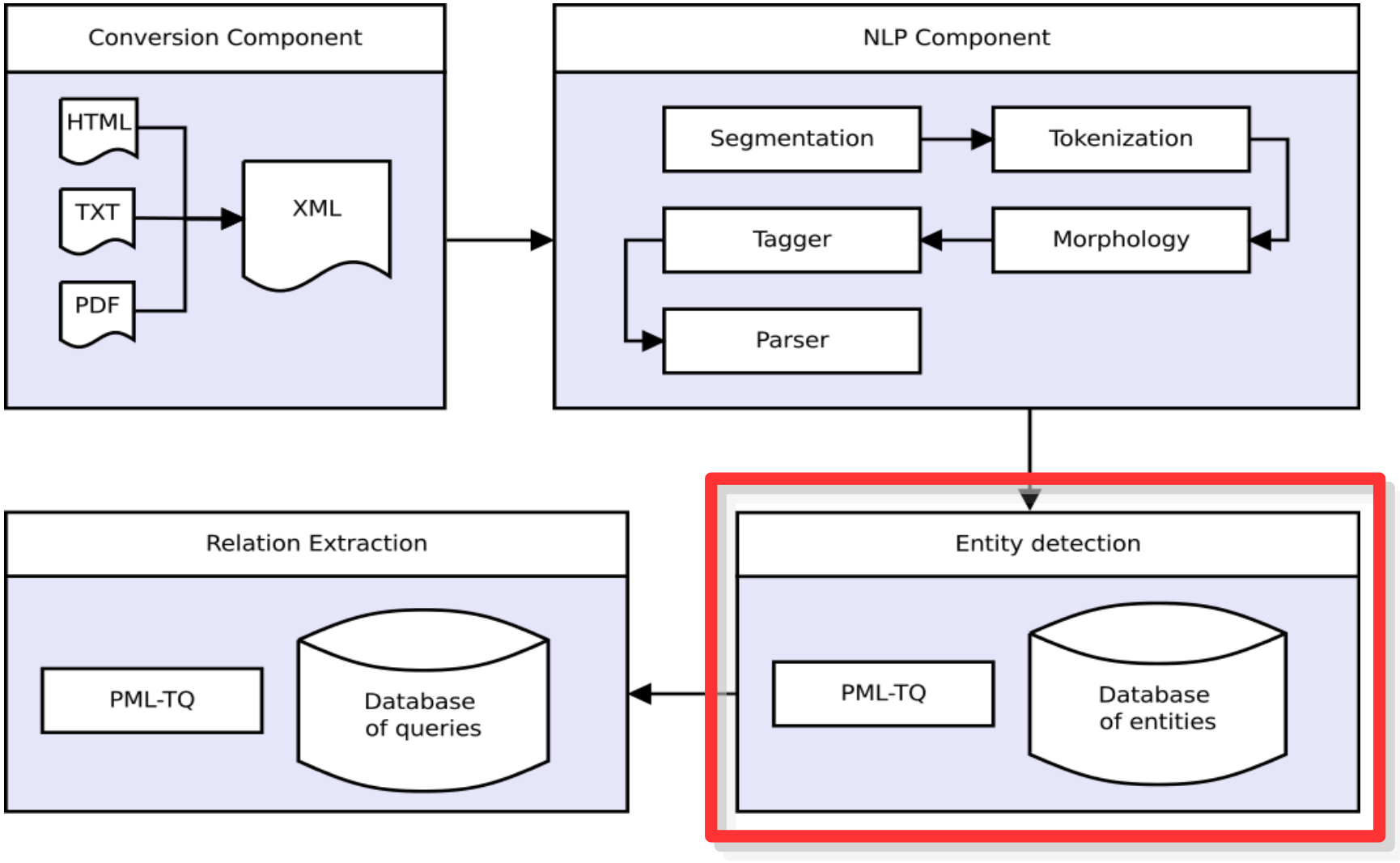
- **Prague Dependency Treebank** framework
  - http://ufal.mff.cuni.cz/pdt3.0
- **Tools**
  - segmentation & tokenization
  - lemmatization & morphology
  - syntactic parsing
  - Treex (http://ufal.mff.cuni.cz/treex)

NLP Component



Accounting units create fixed items and reserves according to special legal regulations

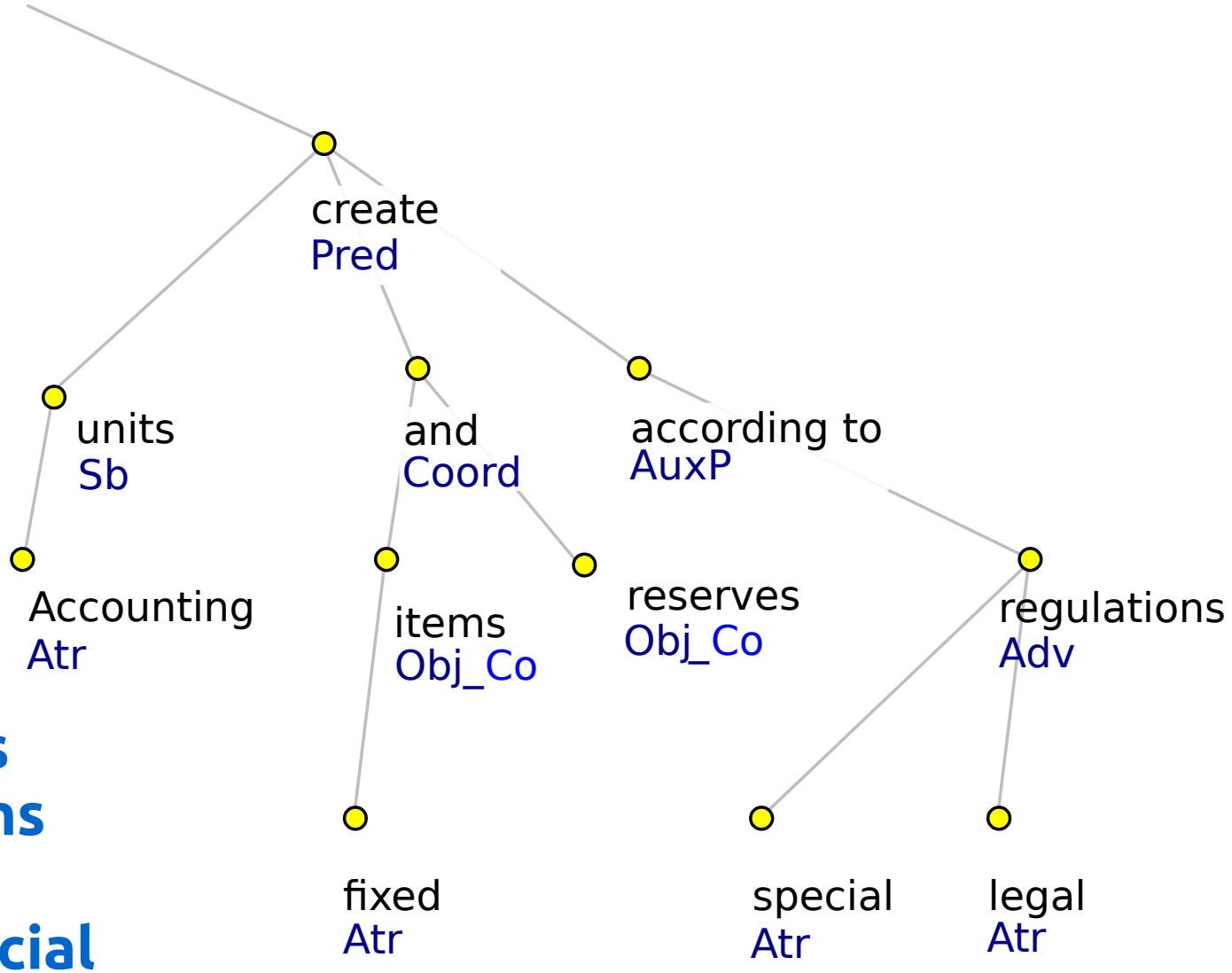
RExtractor Architecture



Entity Detection Component

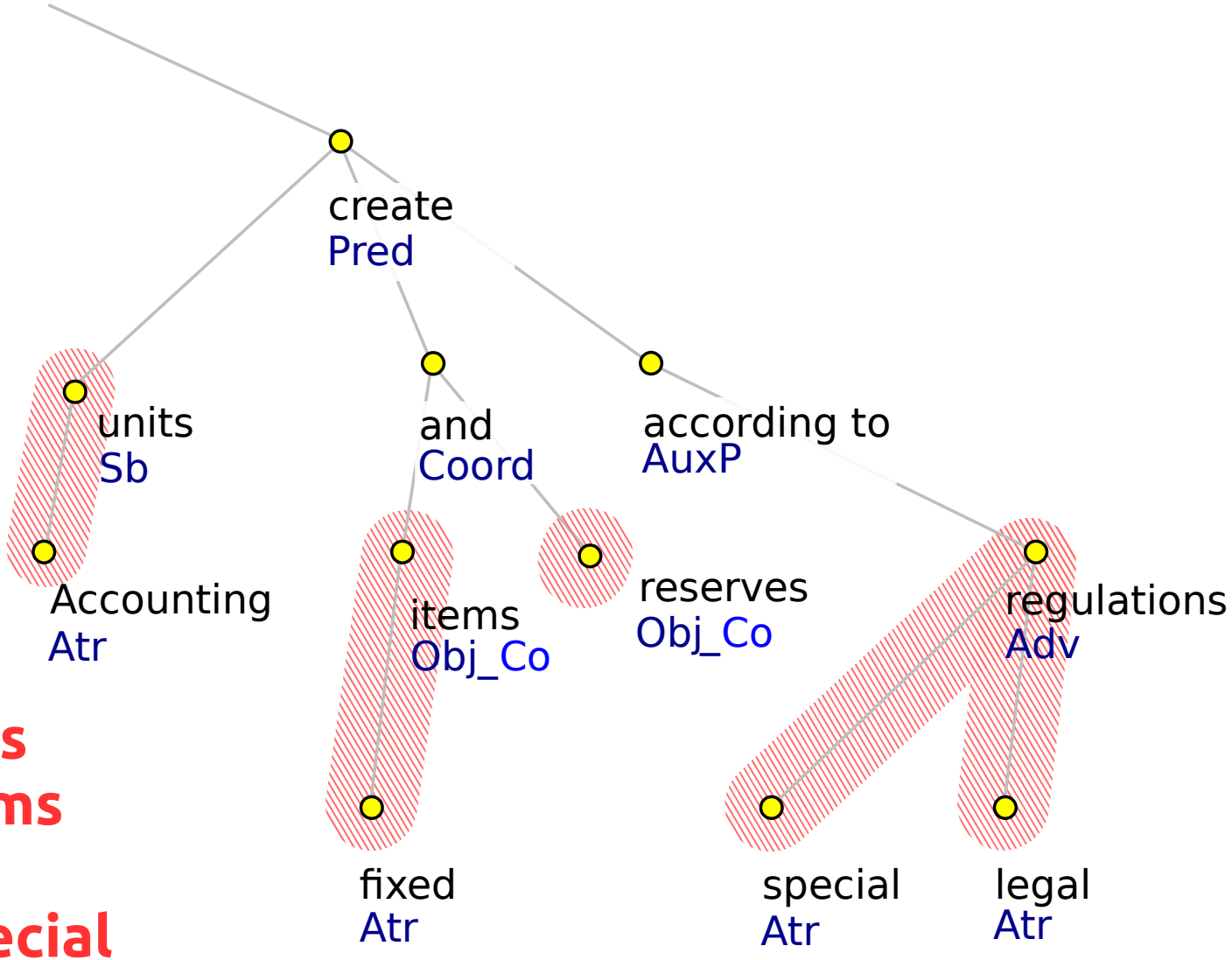
- **Database of Entities**
  - entities specified by domain experts
- **PML-TQ** (http://ufal.mff.cuni.cz/pmltq)
  - tree queries better than regular expressions
    - coordination
    - several word forms in inflective languages

Entity Detection Component



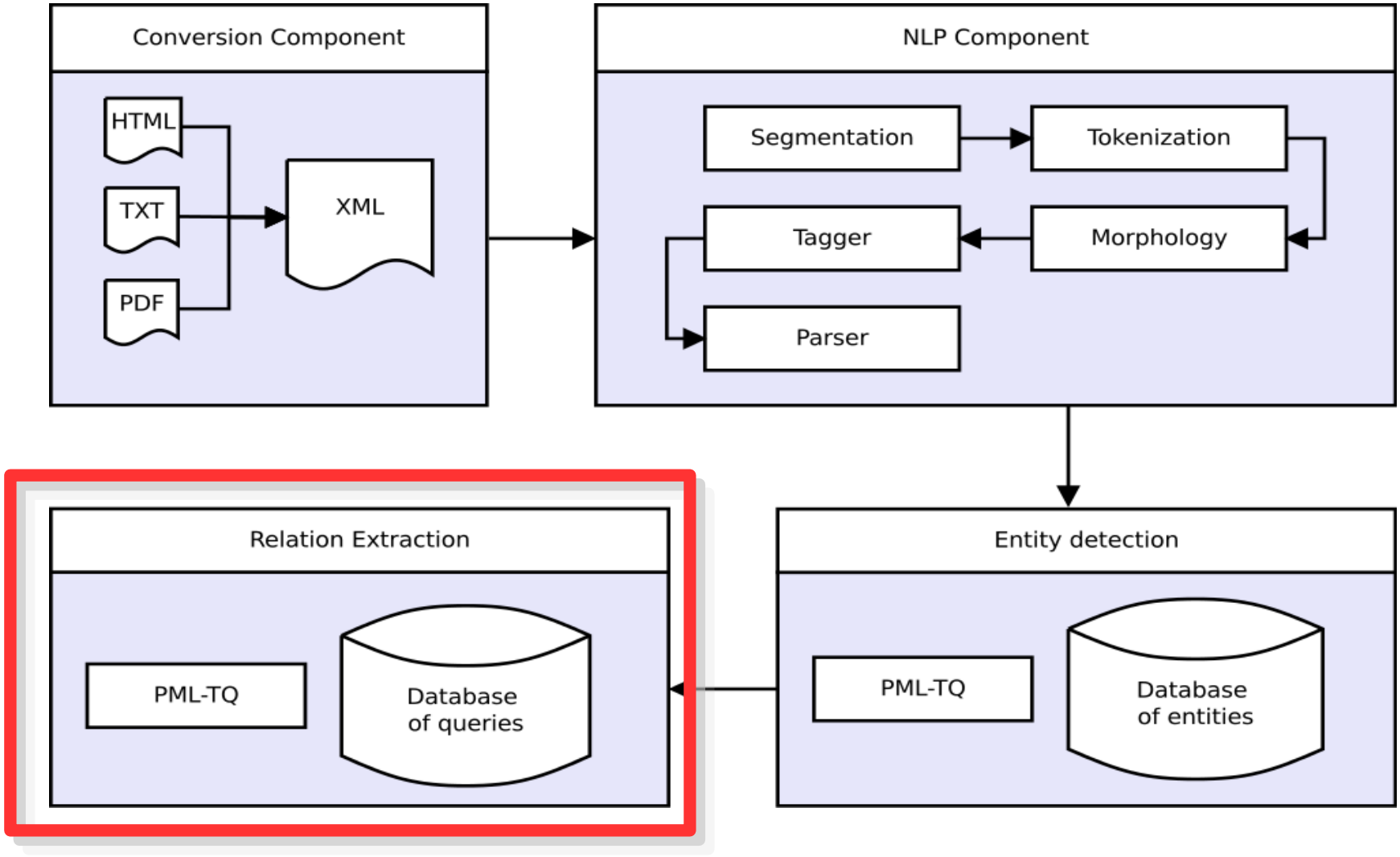
Accounting units create fixed items and reserves according to special legal regulations

Entity Detection Component



Accounting units create fixed items and reserves according to special legal regulations

RExtractor Architecture

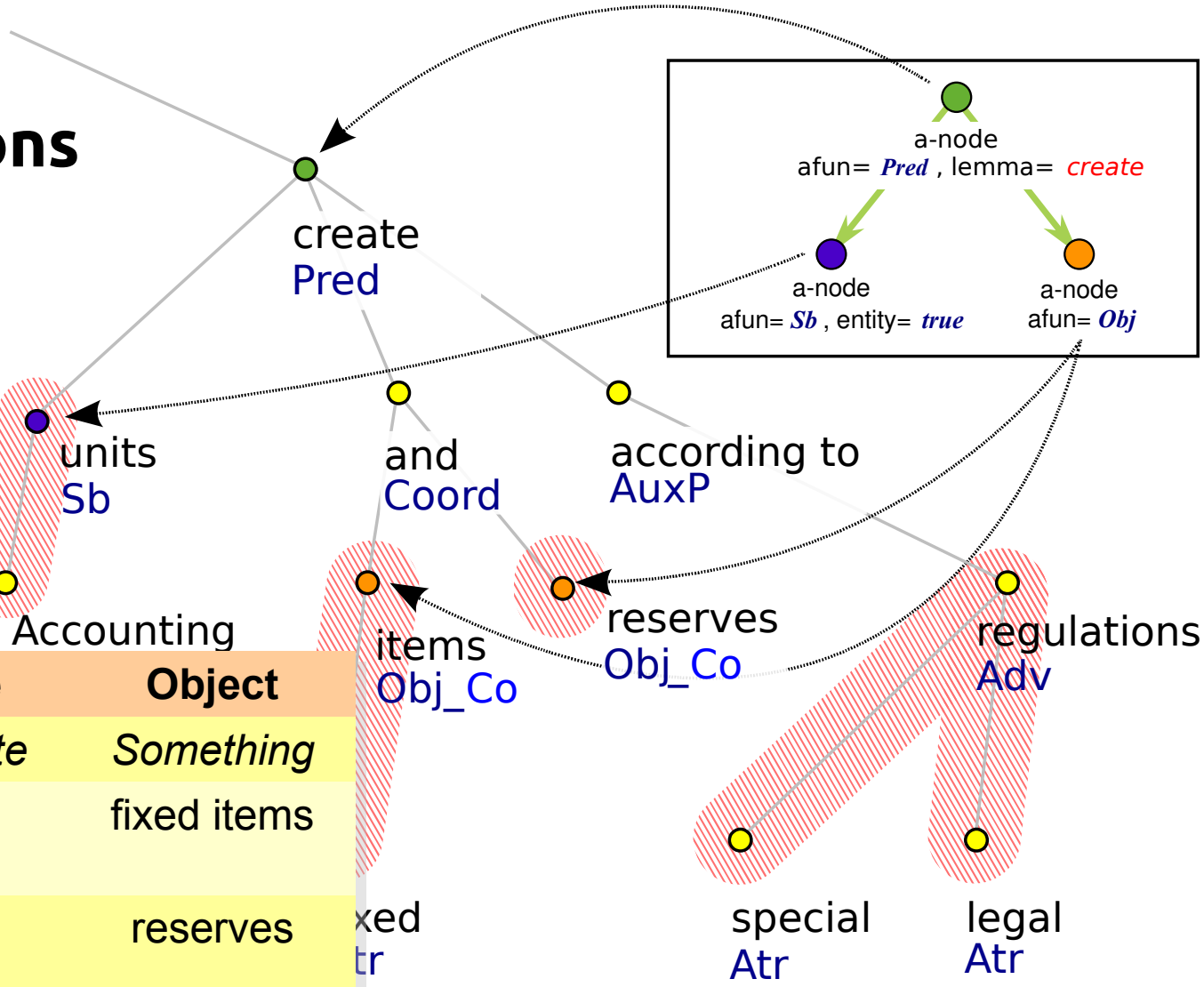


Relation Extraction Component

- **Database of Queries**
  - queries formulated by domain experts
  - their formulation in the form of PML-TQ queries on dependency trees
- **RDF ready output**
  - triples (*subject, predicate, object*)
  - each position
    - is annotated in a text (*text chunk)*
    - has a specific **ontological concept** *(RDF Class)*

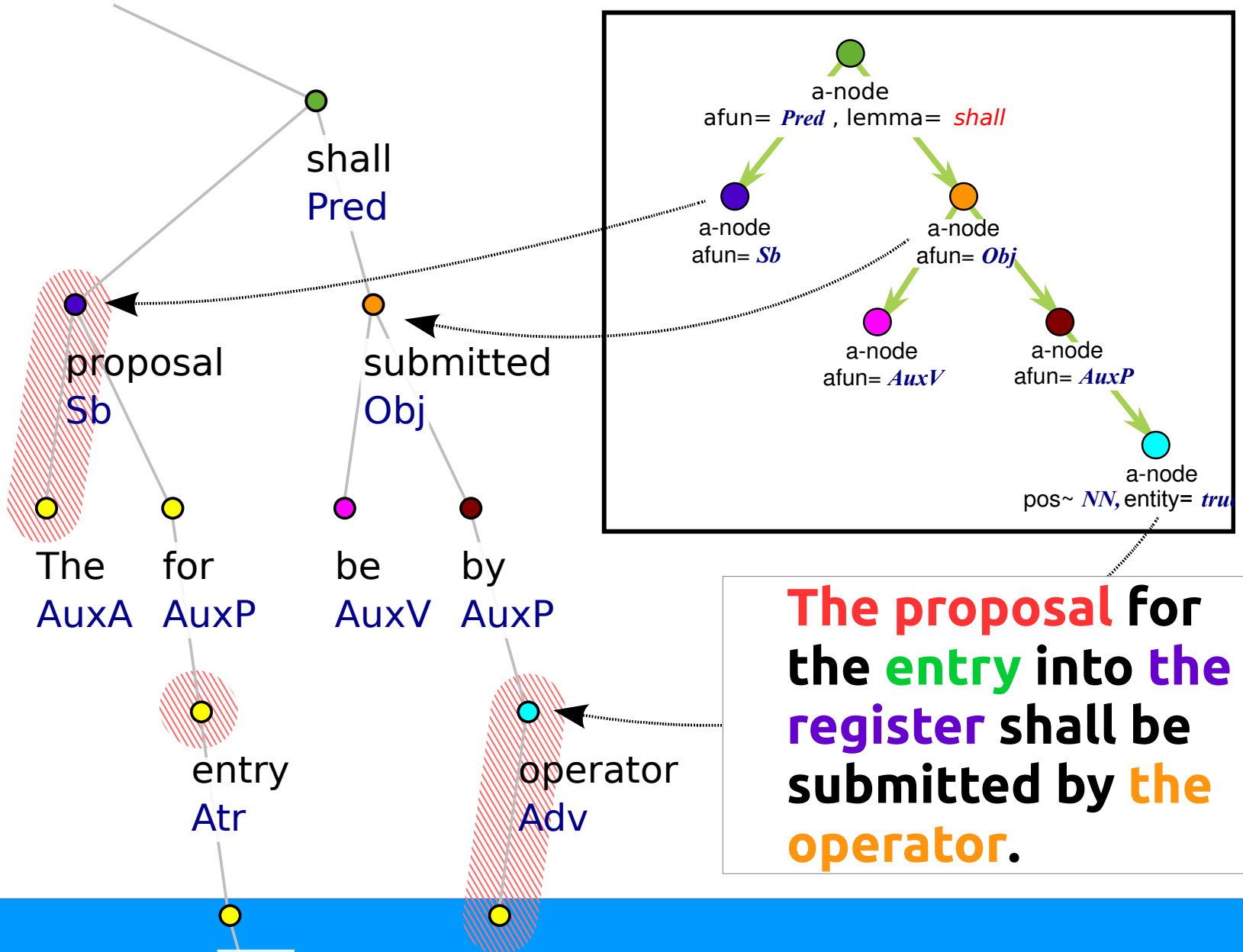
Relation Extraction Component

- **Accounting units' obligations**



Subject	Predicate	Object
Entity	*hasToCreate*	*Something*
Accounting units	create	fixed items
Accounting units	create	reserves

Relation Extraction Component



Relation Extraction Component

Types of relations

- **Definitions** **D**
  - entities are defined or explained
- **Obligations** **O**
  - an entity is obligated to do something
- **Rights** **R**
  - an entity has right to do something

Relation Extraction Component

Results

	D	R	O	Total
# of queries	5	4	2	11
Goldstandard	97	308	62	467
Extracted	70	255	41	366
True positive	53	206	36	295
False negative	44	102	26	172
False positive	17	49	5	71
Precision (%)	**75.7**	**80.8**	**87.8**	**80.6**
Recall (%)	**54.6**	**66.9**	**58.1**	**63.2**

Relation Extraction Component

Error analysis

Error	# of errors	Ratio
Parser	145	59.7%
Query	93	38.3%
Entity	5	2.1%

Results

- errors in automatic parsing
- query design

Conclusion

- general pipeline for **extraction** and **representation** of information that is presented in raw texts
  - processes input texts by linguistically-aware tools
  - extracts entities and relations from dependency trees
  - Linked Data principles
- **Legal documents** as a pilot domain