Vincent Kríž & Barbora Hladká

# RExtractor: a Robust Information Extractor

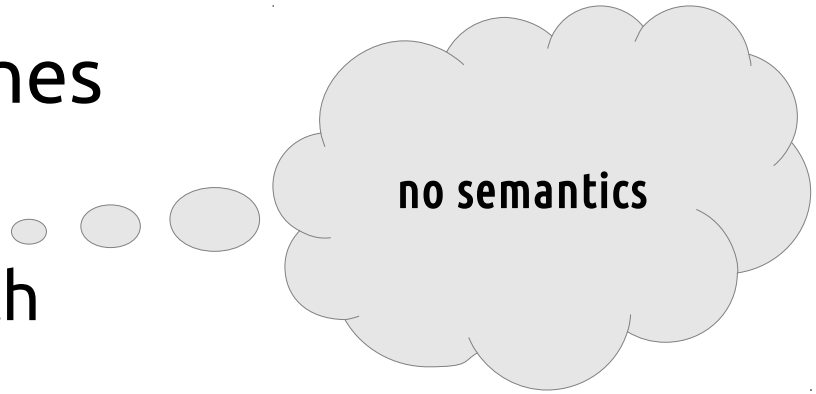
Aplikace NLP, 17.3.2015
MFF UK

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

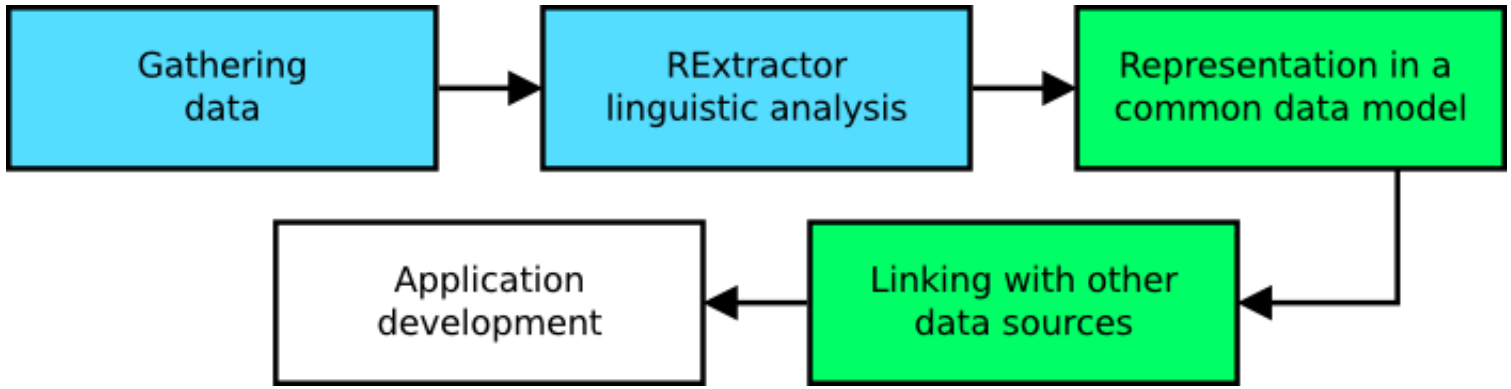
# Motivation

- 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

# Scenario




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

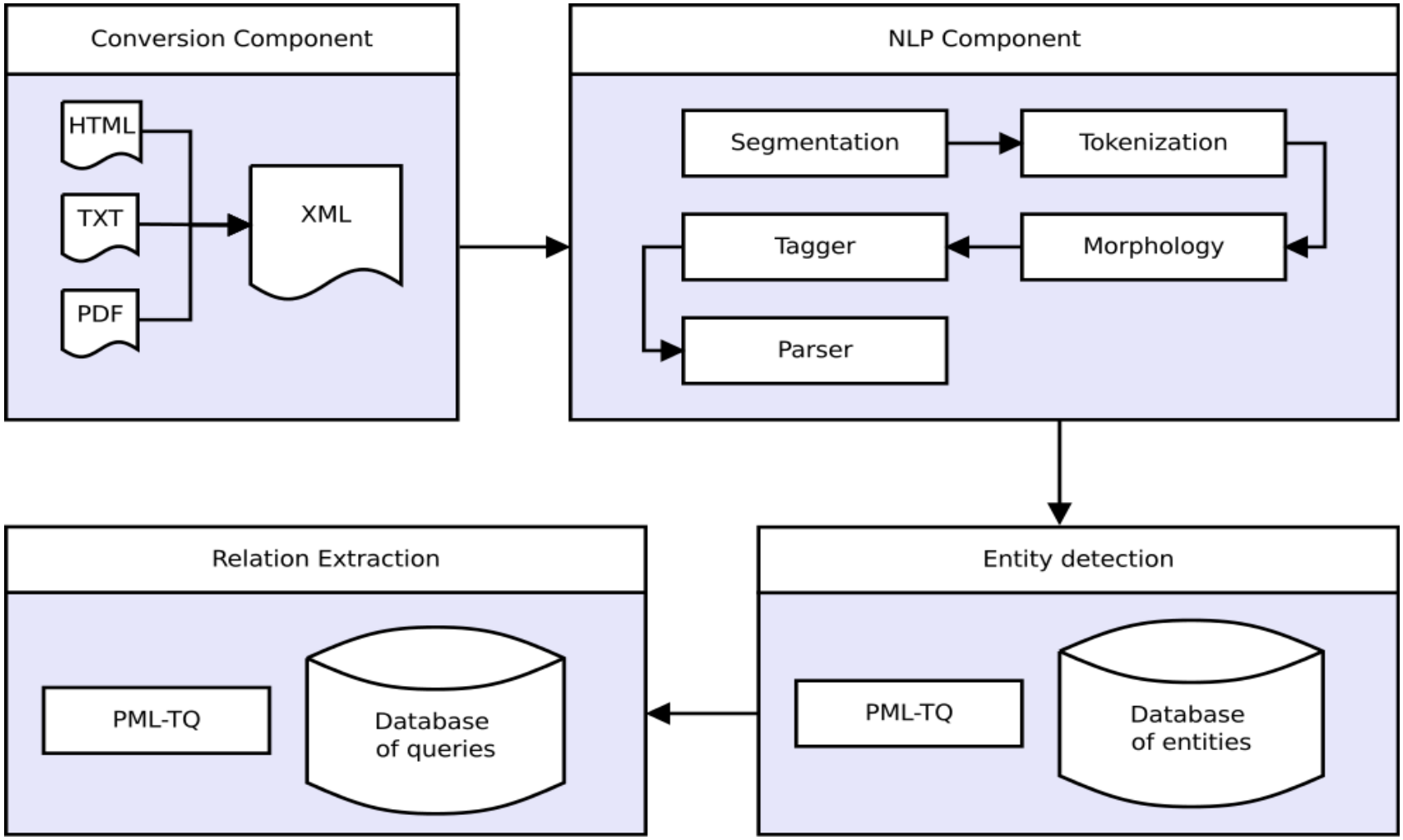
# Scenario

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

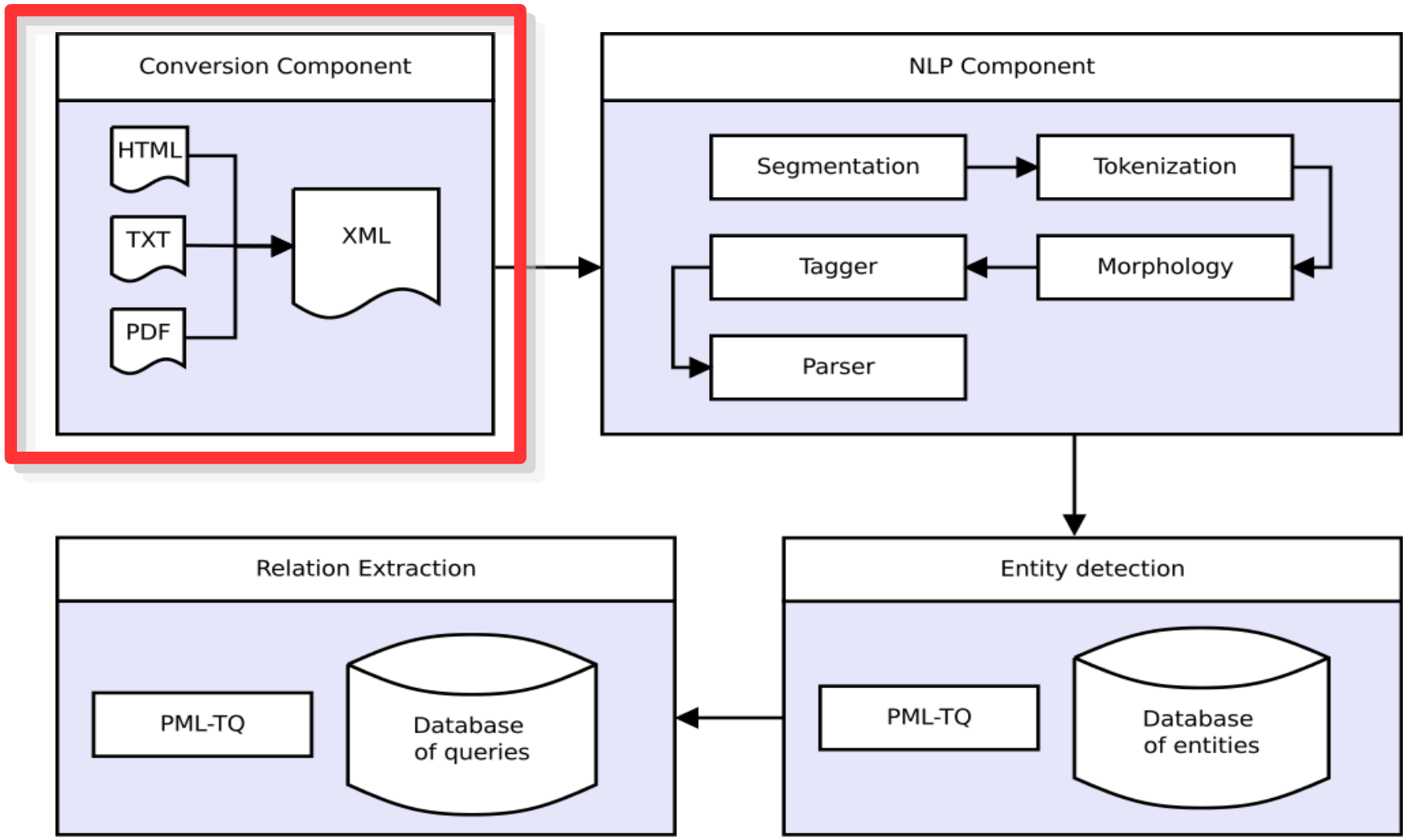
# Scenario

- **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

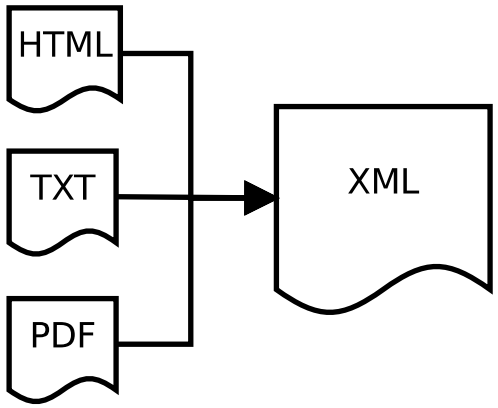


# RExtractor Architecture

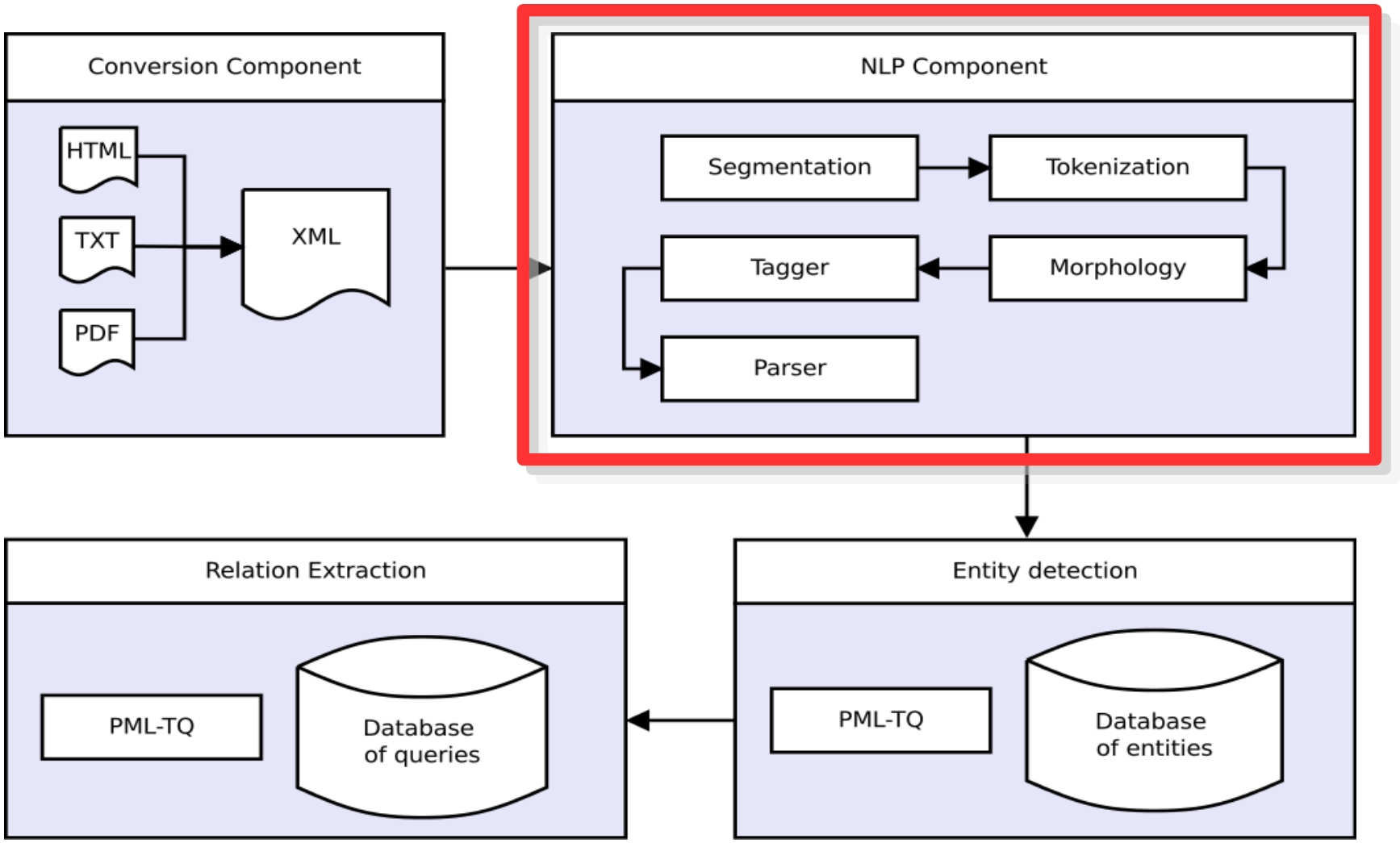


# Conversion Component

- converts various input formats into a unified representation (XML)



# RExtractor Architecture

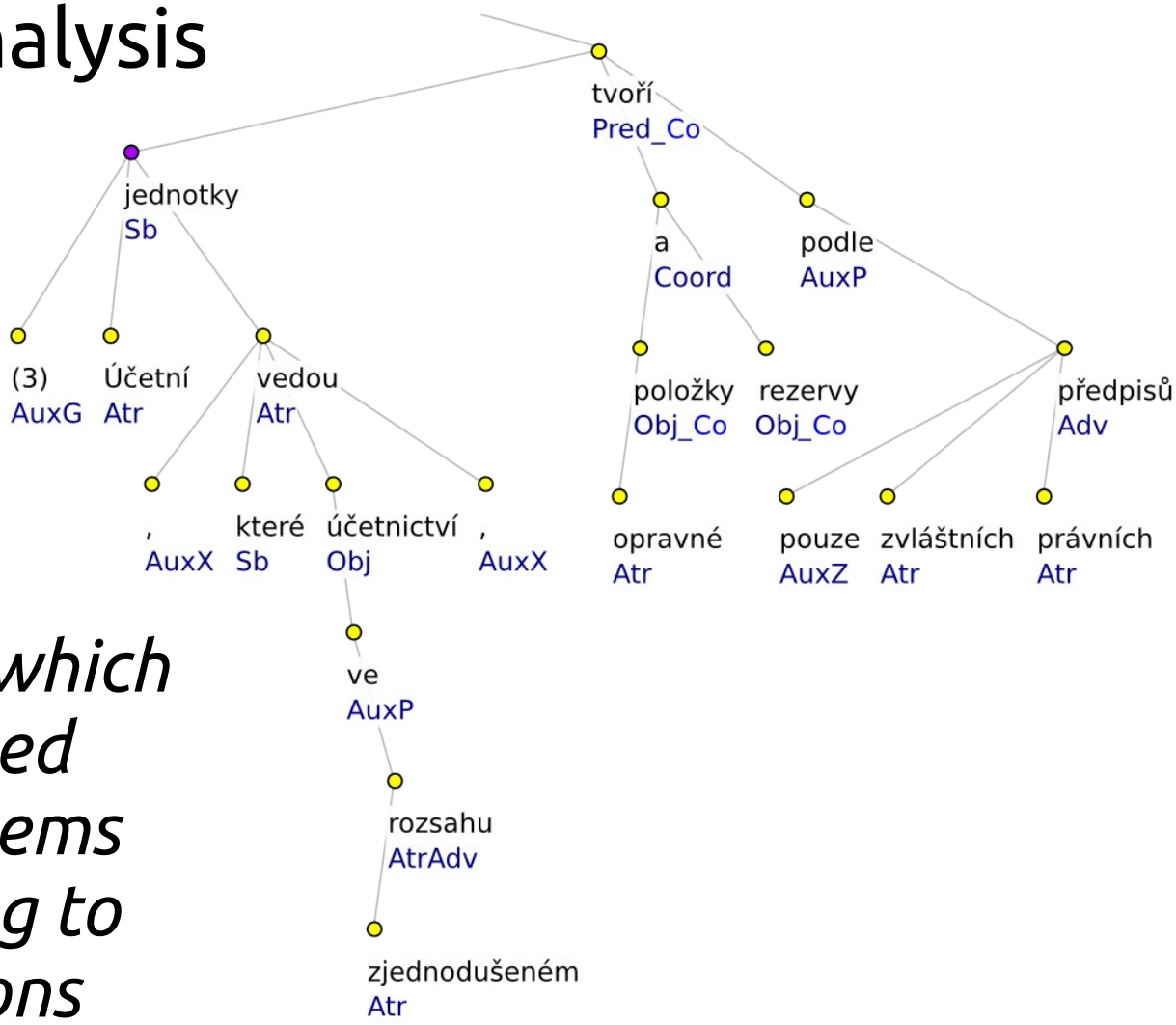


# NLP Component

- **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

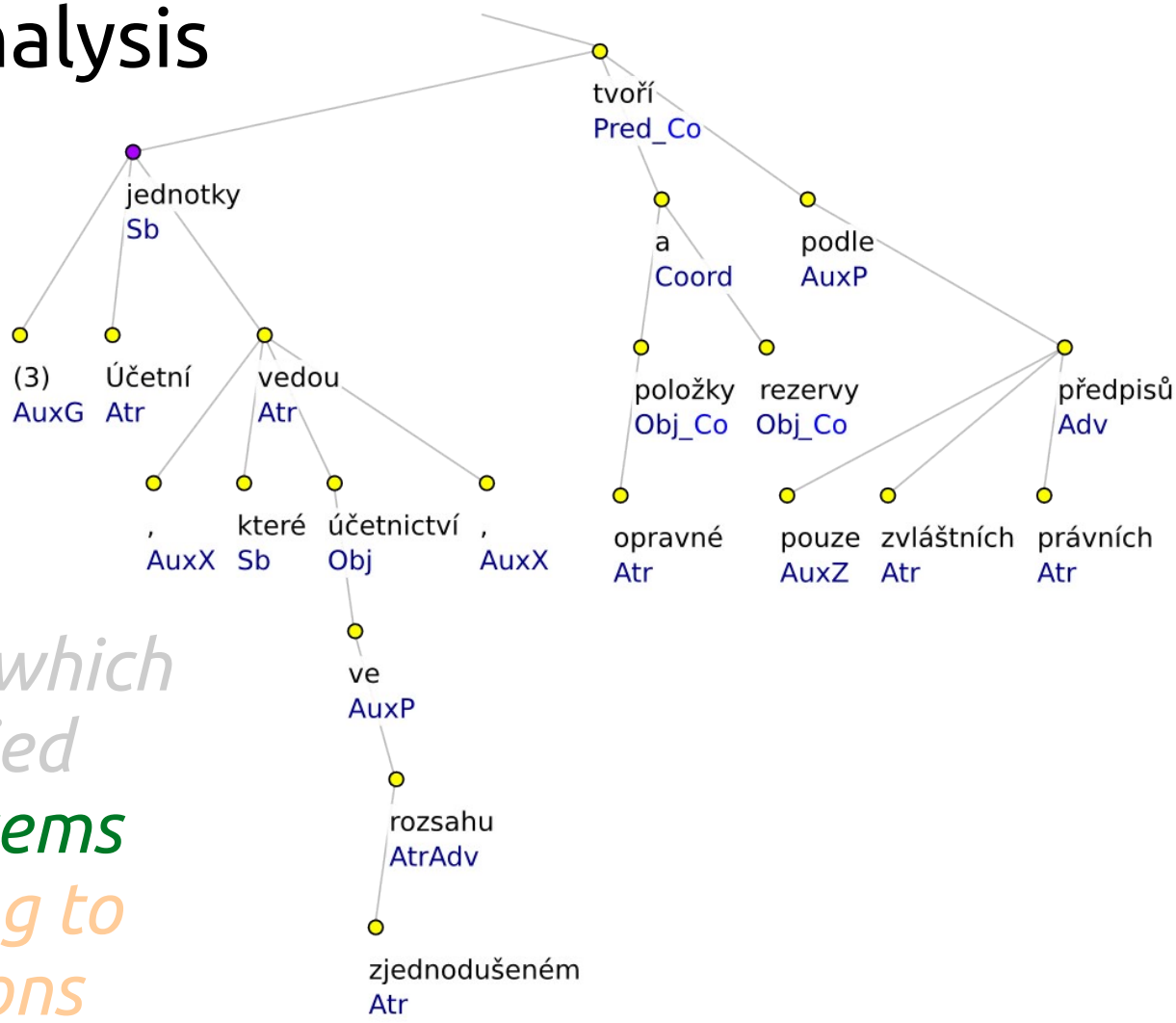
- Output of the analysis




*(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations*

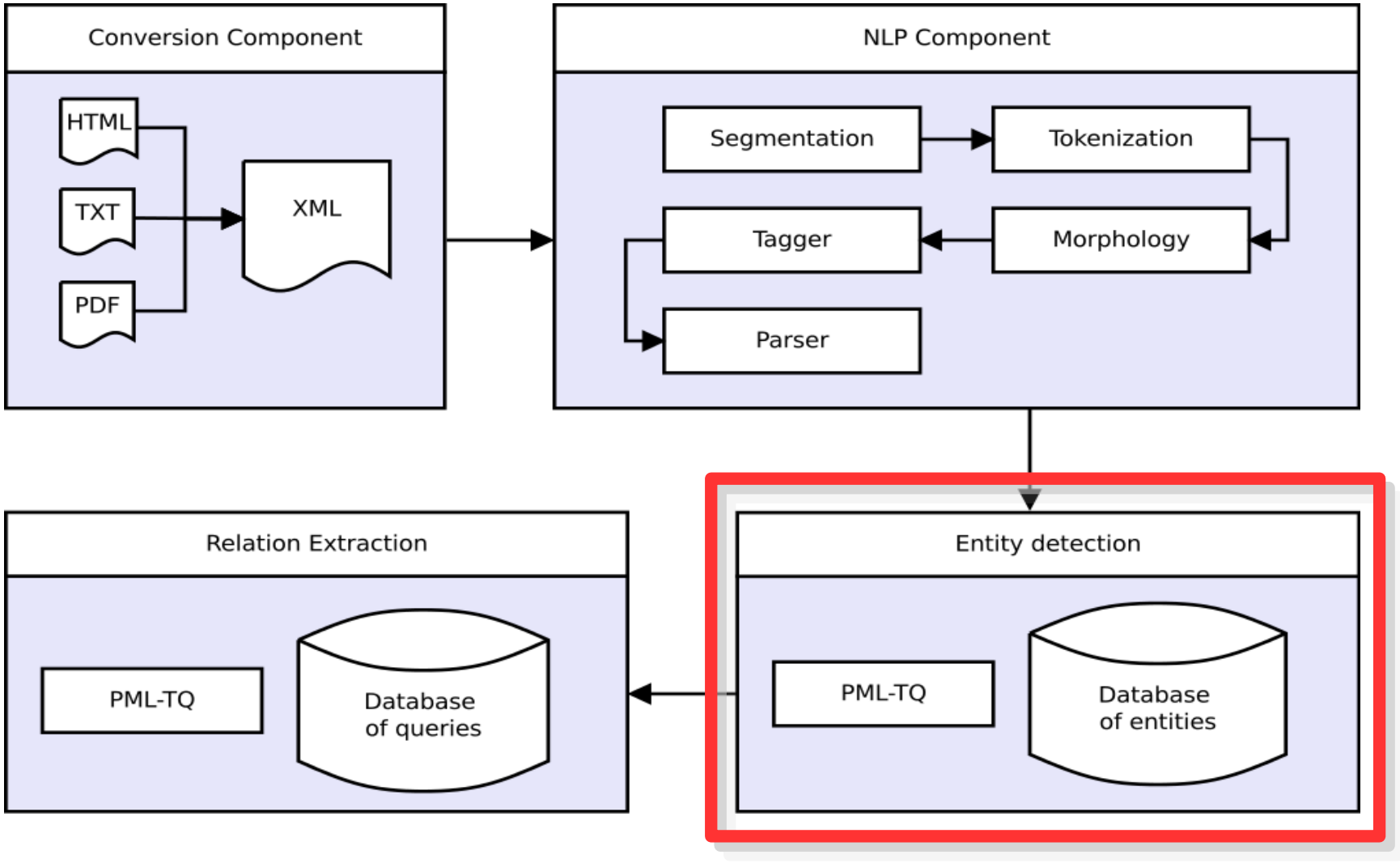
# NLP Component

- Output of the analysis




*(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations*

# RExtractor Architecture

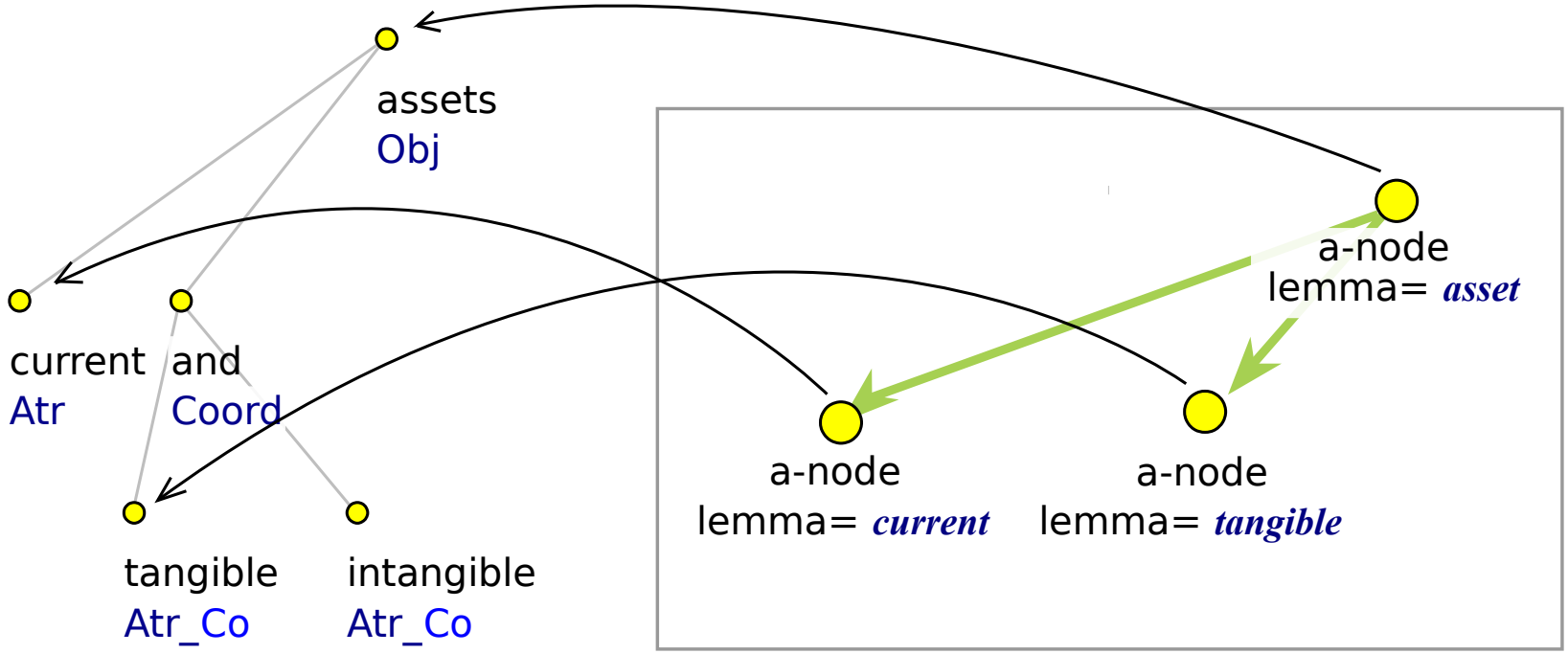


# Entity Detection Component

- **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
  - find the entity *current tangible assets* in the text *current tangible and intangible assets*

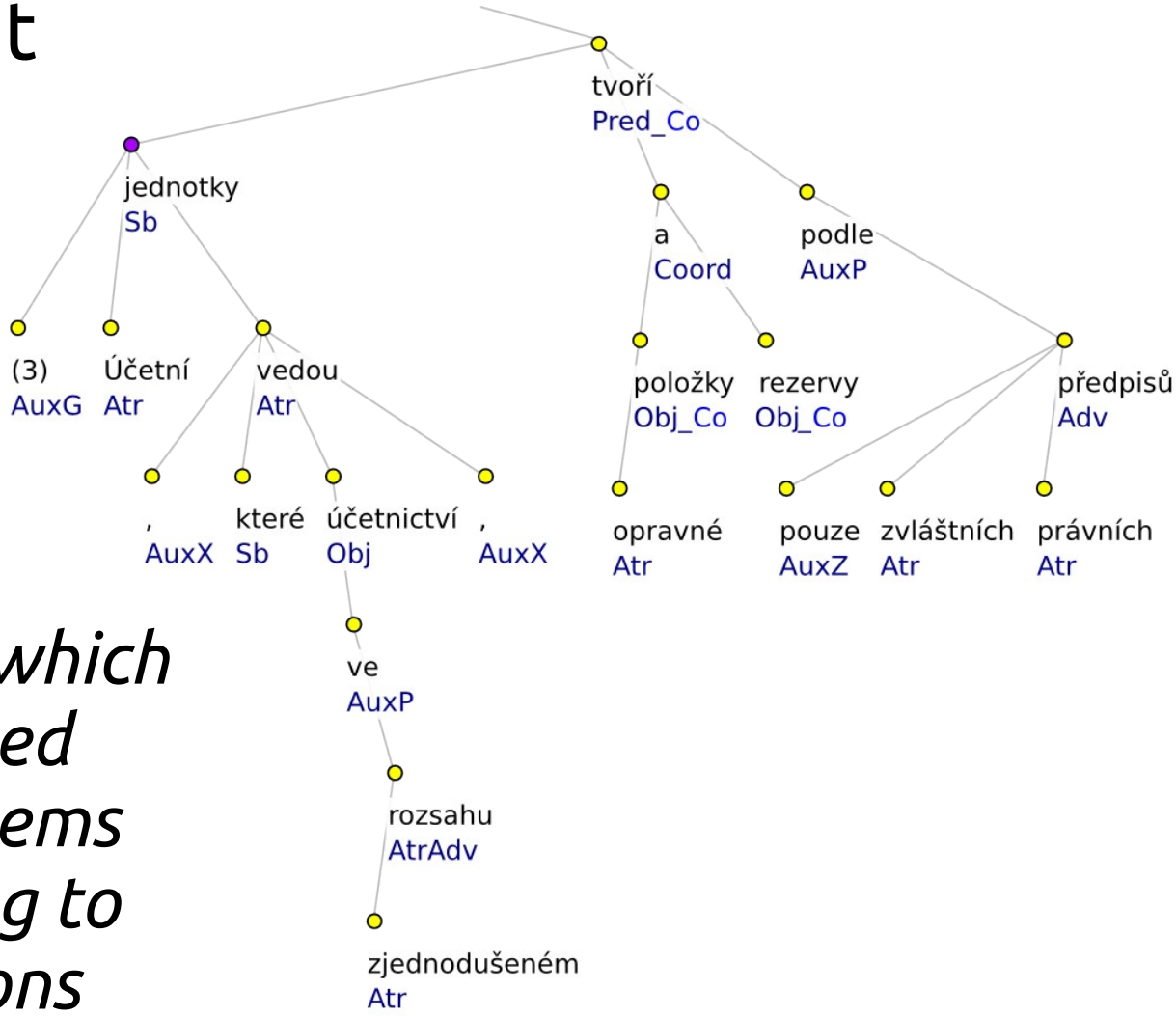
# Entity Detection Component

- find the entity *current tangible assets* in the text *current tangible and intangible assets*



# Entity Detection Component

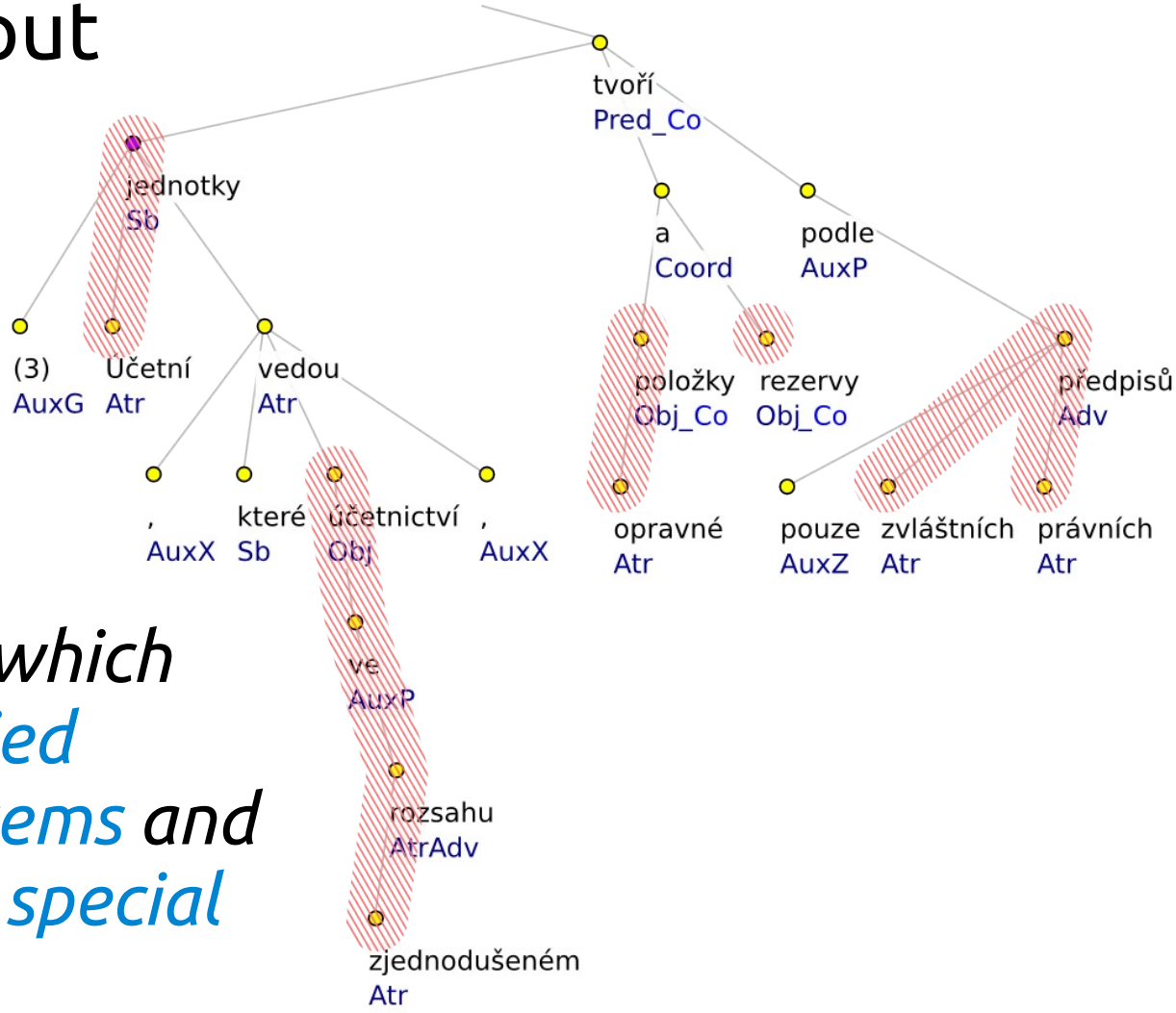
- Component input




*(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations*

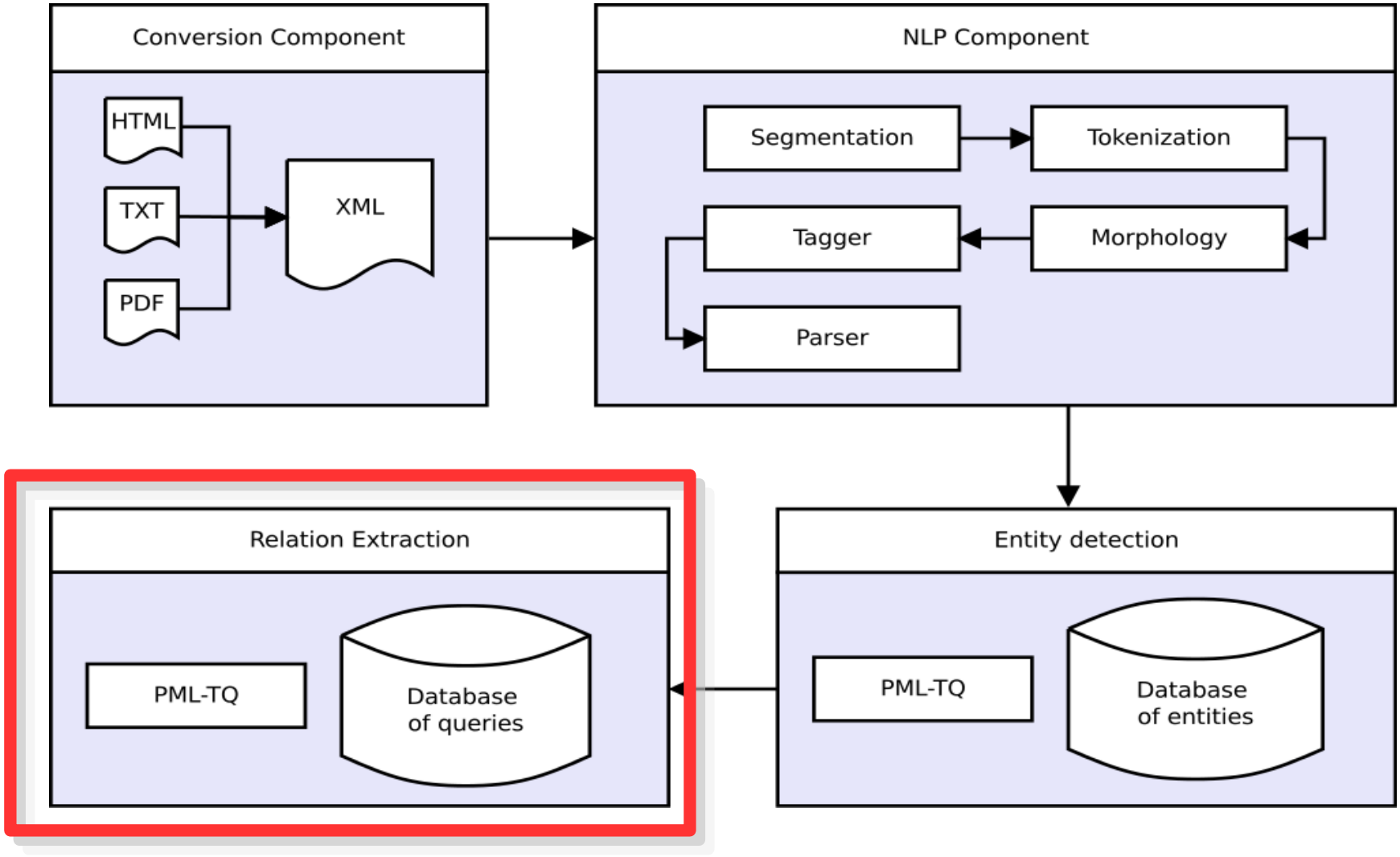
# Entity Detection Component

- Component output




*(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations*

# RExtractor Architecture

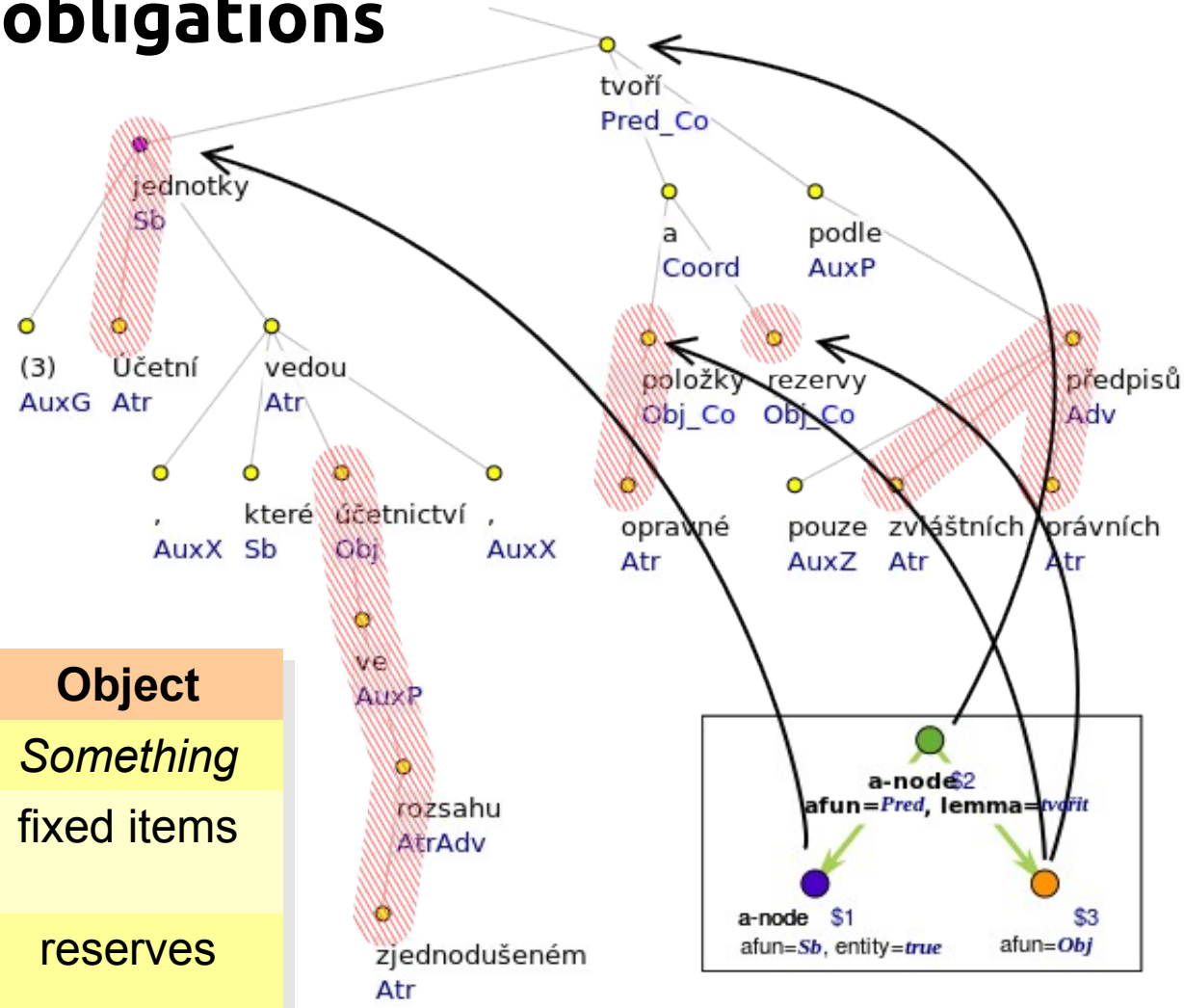


# Relation Extraction Component

- **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 |

# Case study on legislative domain

## Legal texts

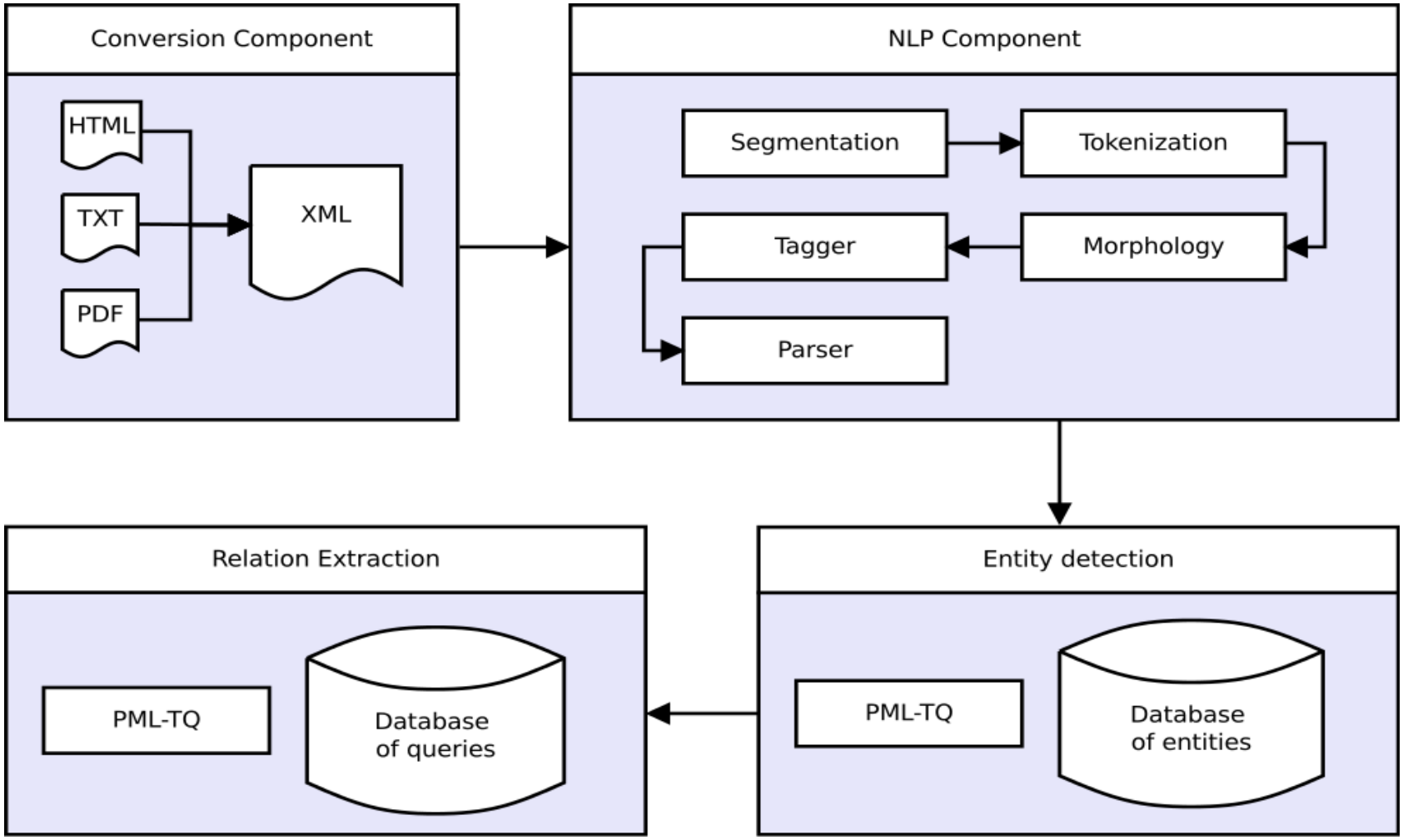
- specialized texts operating in legal settings
- they should transmit legal norms to their recipients
- they need to be clear, explicit and precise

## Sentences

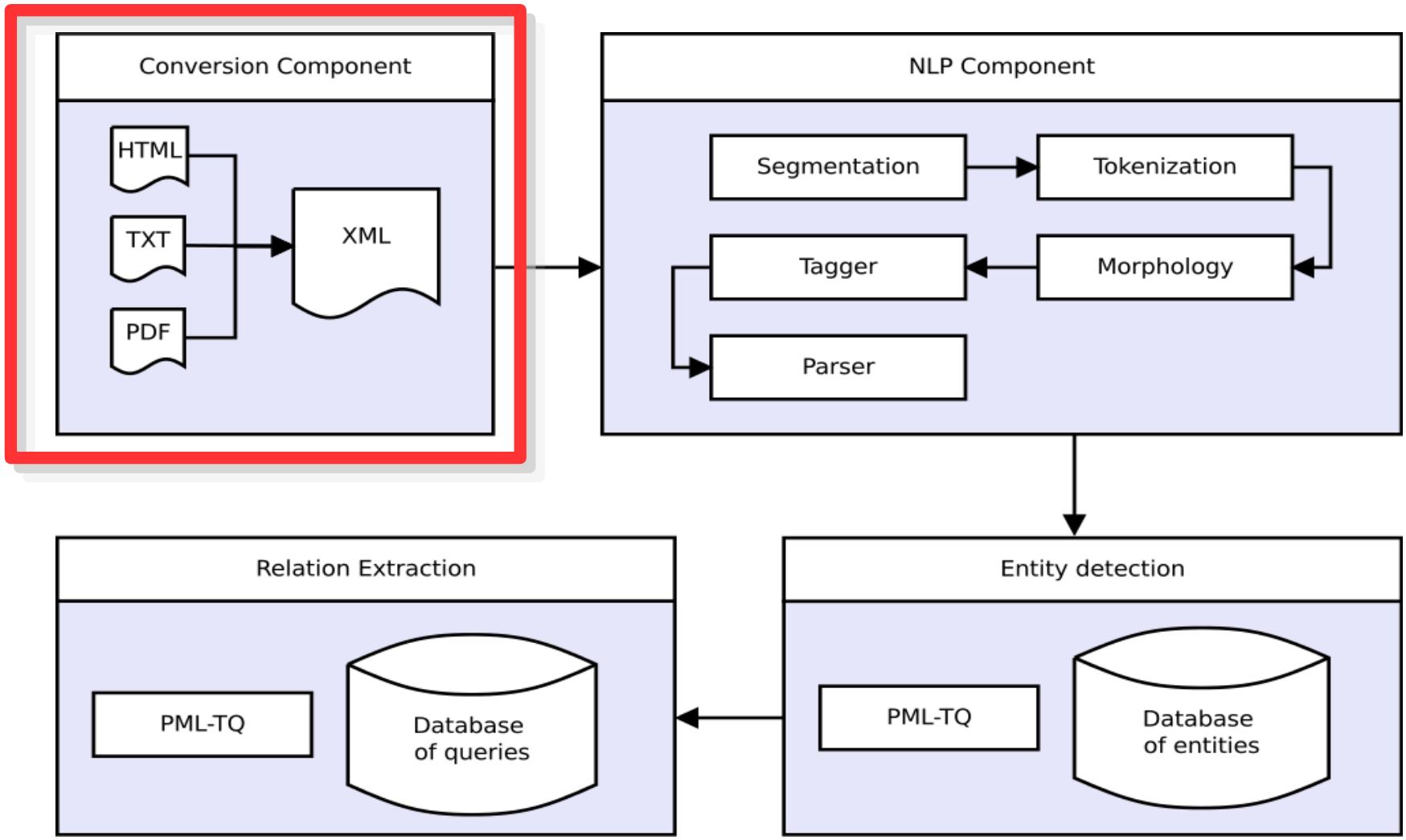
- simple sentences are very rare
- usually long and very complex

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

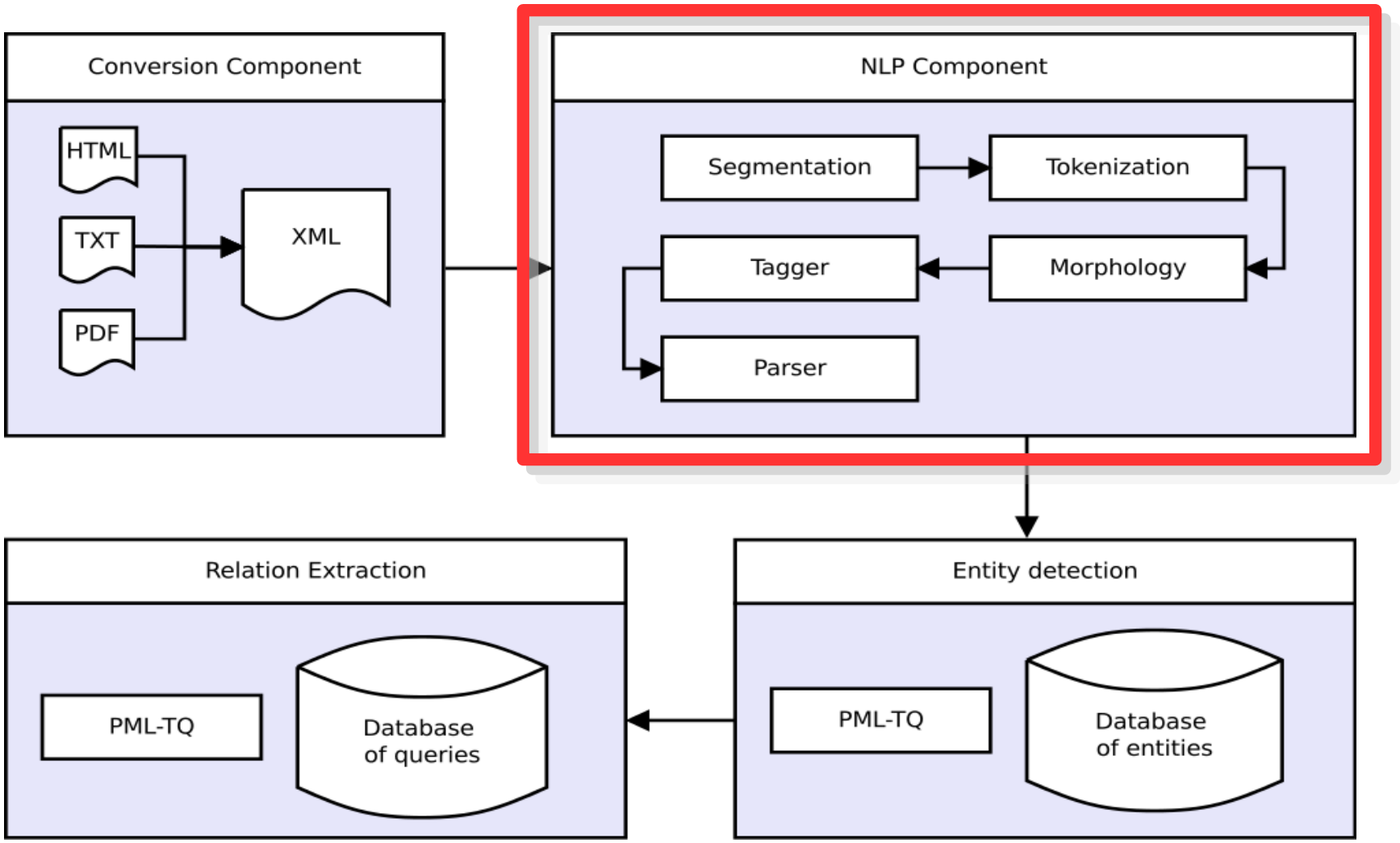
# RExtractor Architecture



# RExtractor Architecture



# RExtractor Architecture



# NLP Component

## Automatic parsers for Czech

- trained on **newspaper texts**
- verification whether we can use the parser trained on newspaper texts or some modifications are needed
- **MST parser**
  - Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Hajič (2005): Non-projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of HLT/EMNLP, Vancouver, British Columbia.

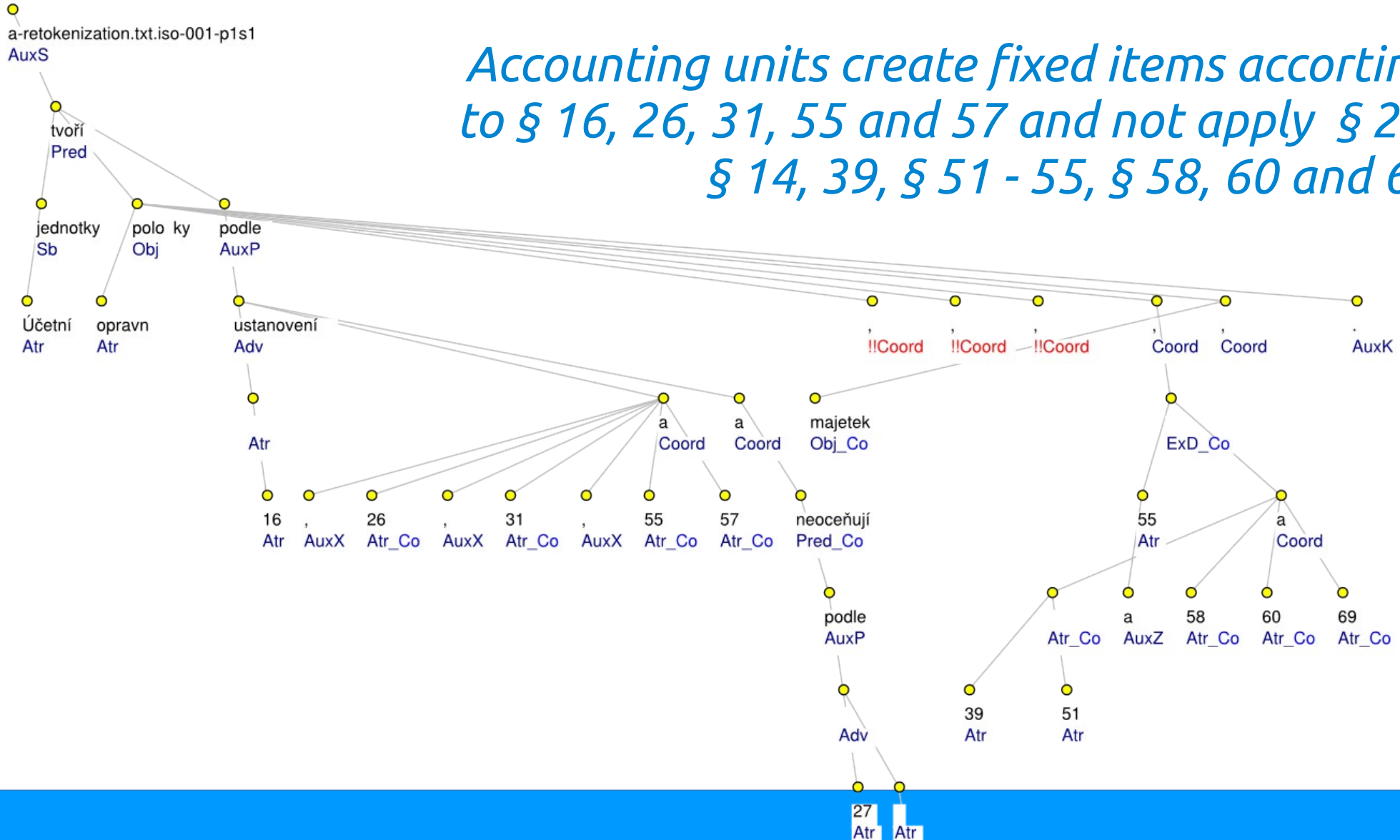
# NLP Component

## Corpus of Czech legal texts (CCLT)

- Accounting Act (563/1991 Coll.)
- Decree on Double-entry Accounting for undertakers (500/2002 Coll.)
- automatically parsed, then manually checked
  - 1,133 manually annotated dependency trees
  - 35,085 tokens

# NLP Component

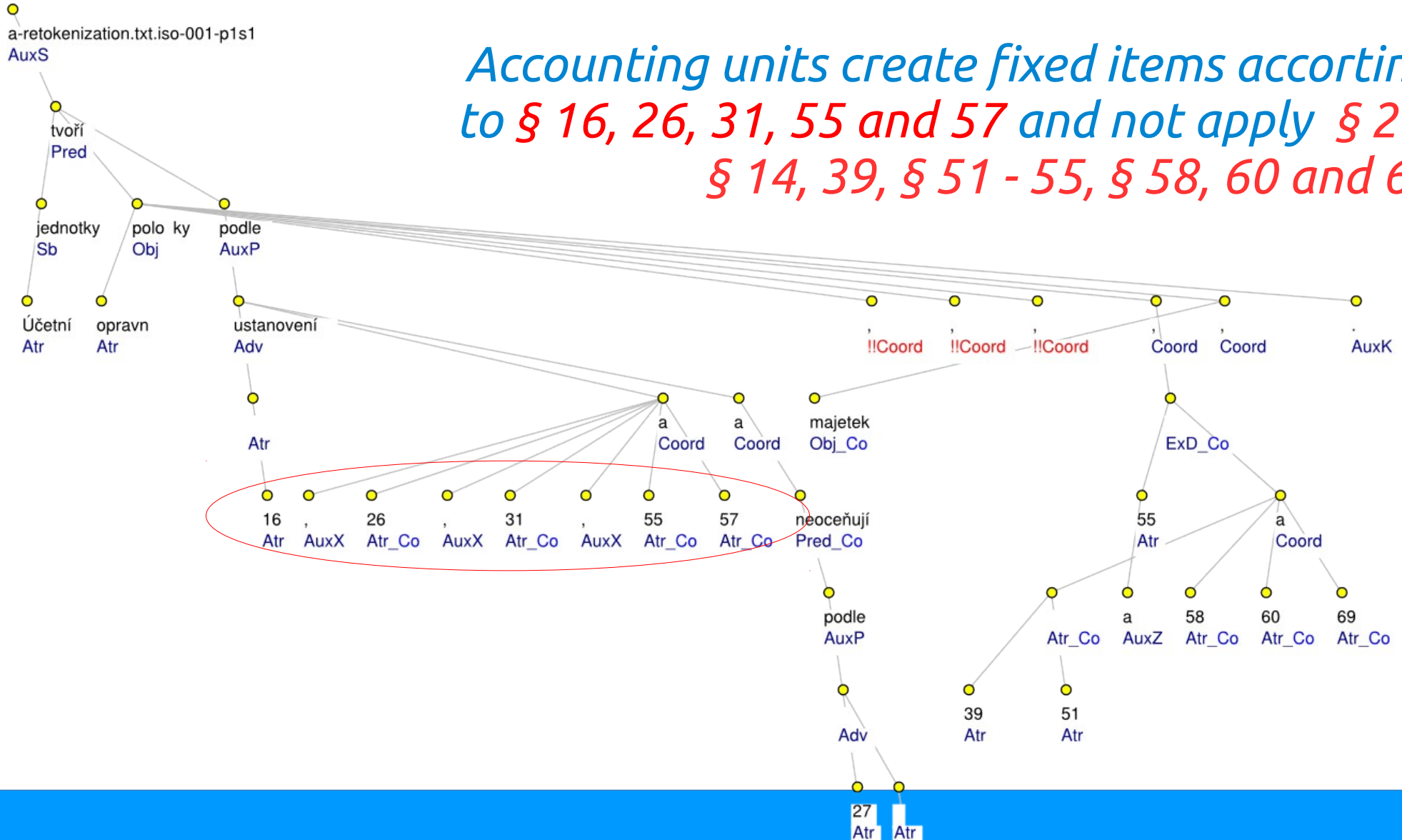
## Re-tokenization




*Accounting units create fixed items accorting to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69*

# NLP Component

## Re-tokenization

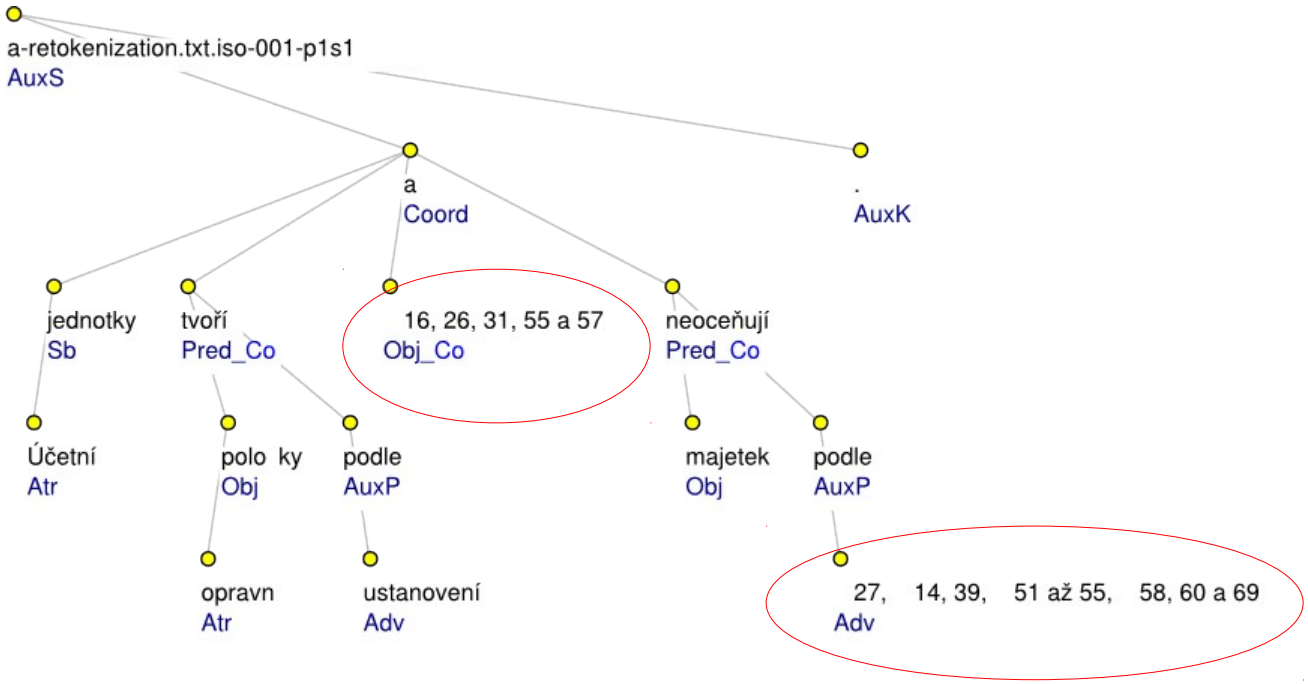



*Accounting units create fixed items accorting to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69*

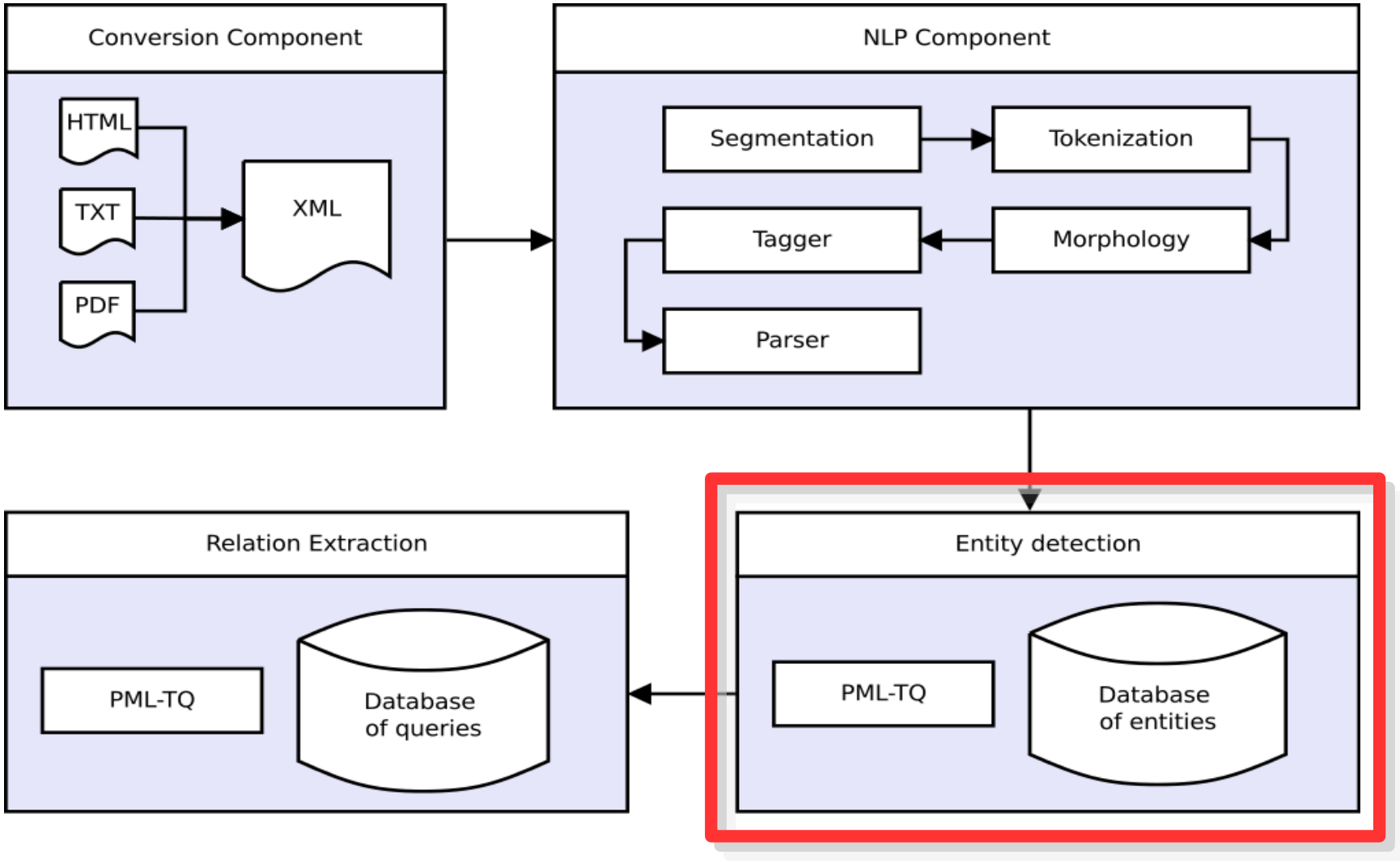
# NLP Component

## Re-tokenization

*Accounting units create fixed items accorting to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69*



# RExtractor Architecture



# Entity Detection Component

## Entities in CCLT

- Accounting subdomain
- Entities manually annotated by domain experts
  - Decree on Double-entry Accounting for undertakers (500/2002 Coll.)

## Sample

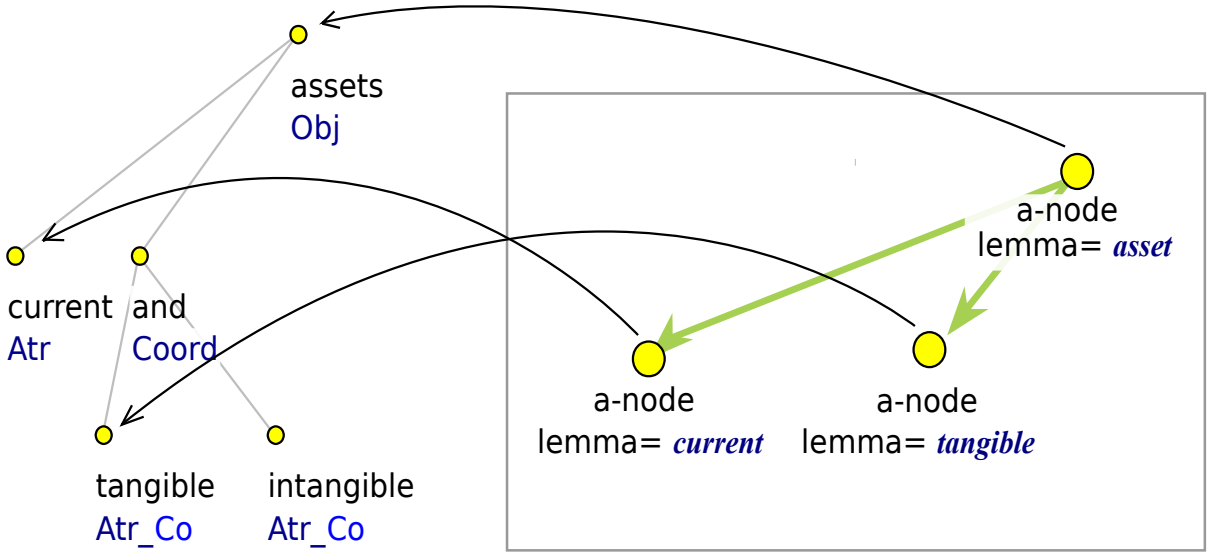
___(1) Vyhláška se vztahuje na účetní jednotky podle § 1 odst. 2 písm. a) a b) zákona, s výjimkou účetních jednotek uvedených v odstavci 2, a na účetní jednotky podle § 1 odst. 2 písm. d) až h) zákona.

___(2) Z účetních jednotek uvedených v odstavci 1 se tato vyhláška nevztahuje na účetní jednotky podle § 19a zákona, pokud zvláštní právní předpis 1c) nestanoví jinak, a na účetní jednotky, jejichž účetnictví upravuje zvláštní právní předpis 1d). Dále se tato vyhláška, s výjimkou § 62 odst. 2 až 5, nevztahuje na účetní jednotky podle § 23a zákona.

# Entity Detection Component

## Initializing DBE with entities from CCLT

- Each (unique) entity parsed automatically by MST
- Automatic procedure takes an entity dependency tree and creates a PML-TQ query



# Entity Detection Component

## Experiment

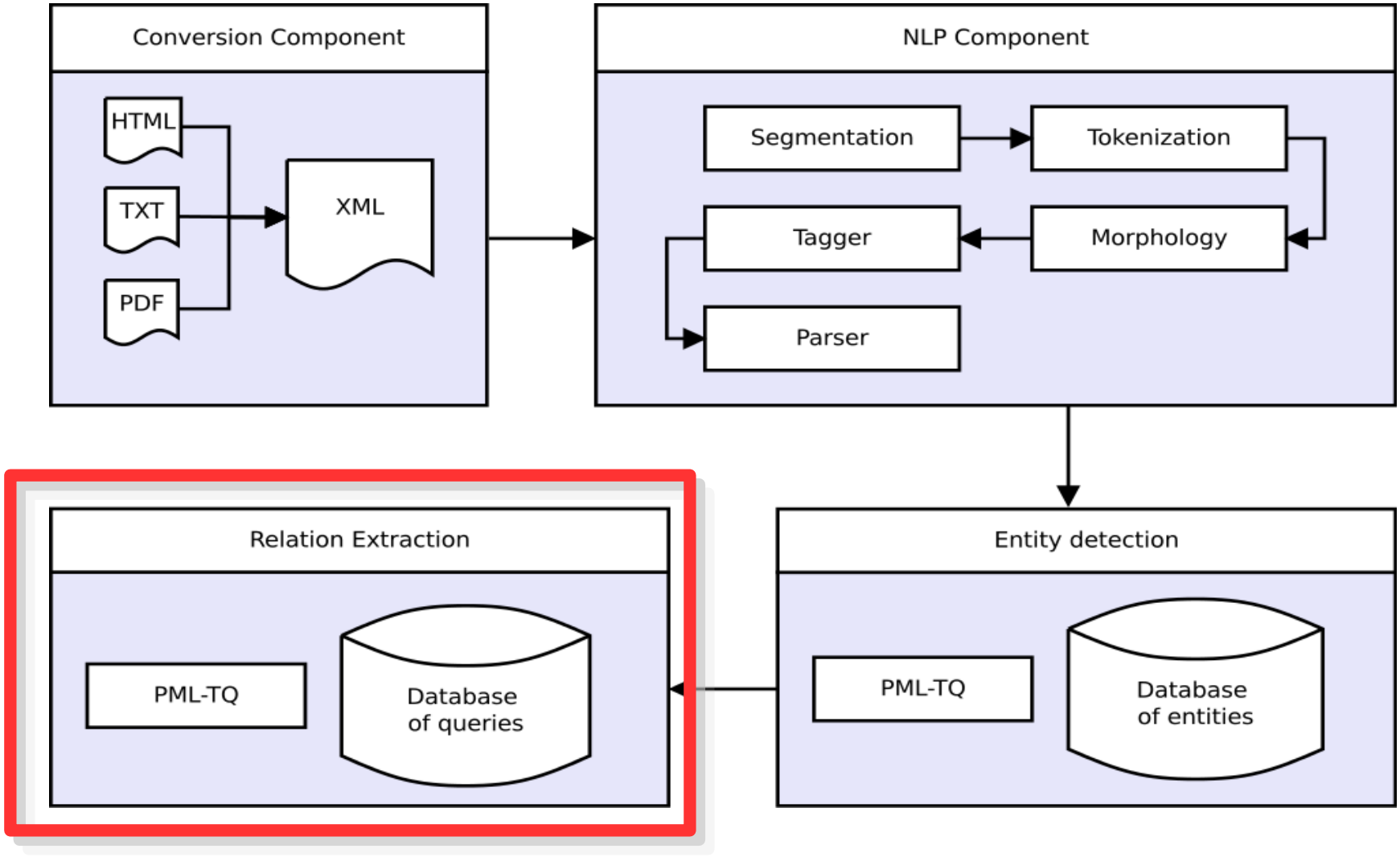
- identify entities in the gold standard trees in CCLT
  - with re-tokenized tokens and *(very) long* sentences
- identify entities in the trees created by MST
  - with re-tokenized tokens and split sentences

| Parsing method | Extracted | TP | FP | FN | Precision | Recall |
|---|---|---|---|---|---|---|
| Manual | 16428 | 9549 | 6879 | 628 | 58.1 | 93.8 |
| Automatic | 16160 | 9278 | 6882 | 838 | 57.4 | 91.7 |

## Results

- high False positives
- automatic parser has low influence on detection

# RExtractor Architecture



# Relation Extraction Component

## Manual design of queries

- Strategy: cover maximum of relations with minimum of queries
- tree query expert
  - observes typical constructions for a given relation type
  - designs a query for the most frequent construction
  - goes through matches and redesigns the query if needed

# 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

**Query design & evaluation on CCLT**

- Query design
  - on *Accounting Act (563/1991 Coll.)*
  - 5 queries for **Definitions**
  - 4 queries for **Rights**
  - 2 queries for **Obligations**
- Evaluation
  - on *Decree on Double-entry Accounting for undertakers (500/2002 Coll.)*

# 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

# Scenario

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

# Legal ontologies

- **Document structure**
  - Act » Expression » Section
- **Document semantic**
  - Legal Concepts Ontology (`lexc:`)
    - represents entities & relations
    - Concept » Concept Version
    - Property » hasDefinition, hasRight, hasObligation, …
  - Linguistic Ontology (`lingv:`)
    - links entities with their appearance in texts

# 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 sentence syntactic representation
  - Linked Data principles
- **Legal documents** as a pilot domain