Ontology-Based XQuery‘ing of XML-Encoded Language Resources on Multiple Annotation Layers

Georg Rehm¹, Richard Eckart², Christian Chiarcos³, Johannes Dellert¹

University of Tübingen¹
SFB 441: Linguistic Data Structures
Tübingen, Germany

TU Darmstadt²
Dept. of English Linguistics
Darmstadt, Germany

University of Potsdam³
SFB 632: Information Structure
Potsdam, Germany

Language Resources and Evaluation Conference – LREC 2008
Context

- Long-term availability of linguistic resources
- Joint Project “Sustainability of Linguistic Data”
- Consolidation of the corpora and data formats
  - Tusnelda SFB 441 “Linguistic Data Structures”
  - Exmaralda SFB 538 “Multilingualism”
  - Paula SFB 632 “Information Structure”
• Sustainability Platform for Linguistic Corpora and Resources
  - ~60 highly heterogeneous linguistic resources
• Goals
  - Centralized corpus platform
  - Homogeneous means of accessing and querying
  - Generalisation over
    ▪ Format (Tusnelda, Exmaralda, etc.)
    ▪ Semantics (various tag-sets)
  - Web-based user interface
    ▪ Intuitively usable for linguists
Linguistic Corpora

- Corpus specific queries

---

Query\textsubscript{1} \rightarrow \text{Corpus}\textsubscript{1} \text{TEI}
Query\textsubscript{2} \rightarrow \text{Corpus}\textsubscript{2} \text{Exmaralda}
Query\textsubscript{3} \rightarrow \text{Corpus}\textsubscript{3} \text{Tusnelda}
Query\textsubscript{4} \rightarrow \text{Corpus}\textsubscript{4} \text{XCES}
Query\textsubscript{n} \rightarrow \text{Corpus}\textsubscript{n} \ldots

status quo
Linguistic Corpora

- Query against SPLICR
- SPLICR generalises over corpora
- Common visualisation/export modules

Browsing
Querying etc.

Visualisation (e.g. SVG)
Export (e.g. ODF)

Corpus\textsubscript{1}  
TEI

Corpus\textsubscript{2}  
Exmaralda

Corpus\textsubscript{3}  
Tusnelda

Corpus\textsubscript{4}  
XCES

Corpus\textsubscript{n}  
...

Ontology-Based XQuery’ing of XML-Encoded Language Resources on Multiple Annotation Layers
Processing and Normalisation of Corpus Data

Manual analysis of annotation schemes and annotation layers results in formalisations as OWL ontologies.

Semi-automatic processing and normalisation on the level of XML-based annotations.

- Corpus₁
  - Annotation scheme z
    - Formal model z (OWL)
    - linking
  - Annotation scheme y
    - Formal model y (OWL)
    - linking
  - Annotation scheme x
    - Formal model x (OWL)
    - linking

- Annotation scheme z
- Annotation scheme y
- Annotation scheme x

- Format x (tag set)
- Format y (tag set)
- Format z (tag set)

- Tool₁: Multi-rooted tree
- Tool₂: Multi-rooted tree
- Tool₃: Multi-rooted tree

- XML database

Ontology-Based XQuery’ing of XML-Encoded Language Resources on Multiple Annotation Layers
Processing and Normalisation of Corpus Data

Manual analysis of annotation schemes and annotation layers results in formalisations as OWL ontologies.

Semi-automatic processing and normalisation on the level of XML-based annotations.

- Corpus_1
  - Annotation scheme z
    - Formal model z (OWL)
    - linking
    - OWL-based reference ontology of linguistic annotations

- Corpus_2
  - Annotation scheme y
    - Formal model y (OWL)
    - linking

- Corpus_3
  - Annotation scheme x
    - Formal model x (OWL)
    - linking

Format x (tag set)
Format y (tag set)
Format z (tag set)

Tool_1
Tool_2
Tool_3

Multi-rooted tree
Multi-rooted tree
Multi-rooted tree

XML database

Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers
Normalising Annotation Format

- Model: multi-rooted trees
- XML-encoded corpora split into multiple layers (trees)
  - One XML file per annotation layer
  - All are identical with regard to their primary data
- Normalizing the XML elements and attributes
  - Tool supported and flexibly configurable (Splitter, Leveler)
- Single layer can be queried with standard XML methods
- Multiple layers cannot be queried with standard methods
  - Introduce custom XQuery functions
Processing and Normalisation of Corpus Data

Manual analysis of annotation schemes and annotation layers results in formalisations as OWL ontologies.

Semi-automatic processing and normalisation on the level of XML-based annotations.

Corpus

Corpus

Corpus

Annotation scheme z

Annotation scheme y

Annotation scheme x

Formal model z (OWL)

Formal model y (OWL)

Formal model x (OWL)

Formal model z (OWL)

Formal model y (OWL)

Formal model x (OWL)

Multi-rooted tree

Multi-rooted tree

Multi-rooted tree

Tool

Tool

Tool

XML database

Annotation schemes

 OWL-based reference ontology of linguistic annotations

formalise
Formalising Annotation Semantics

- Corpora differ in their annotation schemes
- Integrated treatment of heterogeneous resources requires
  - Annotation specifics documented using a formal language
  - Integrated access to resources with different annotations
- Ontology-based approach
  - Ontological formalisation of annotation schemes
  - Standard format (OWL/DL)
  - Supported by several tools (Protégé, Pellet)
OLiA: Ontology of Linguistic Annotations

- Annotation Model
  - Ontological formalization of one particular annotation scheme
- OLiA Reference Model
  - Ontological formalization of reference terminology
- Linking
  - Concepts (and tags) of an annotation model are defined with reference to the OLiA Reference Model
    - Sub-concepts/sub-properties \( \subseteq \in \)
    - Complex expressions \( \n \cup \)
- An example
  - POS tag APPGf “her” [Susanne Tagset]
OLiA: Ontology of Linguistic Annotations
OLiA: Ontology of Linguistic Annotations

• Annotation model
  - 10 models for European and non-European languages
  - POS, morphology, syntactic labels, co-reference, information structure

• OLiA Reference Model
  - Based on terminological references, esp. EAGLES, GOLD

• Linking
  - Extensible architecture
  - Linking with external Reference Models
  - (GOLD, OntoTag, Data Category Registry) supported
Graphical Query Interface Requirements

- Intuitively usable graphical query interface
- Work with multi-rooted trees
- Include the ontology of linguistic annotations into queries
- Work with open standards, i.e., XQuery, OWL
SPLICCR Graphical Query Interface

- SPLICCR has an intuitive graphical query interface
- Generalises over the underlying data structures and querying
- Tree fragment query editor
  - Ontology-supported abstraction of linguistic concepts
  - Operands glue together concepts to construct complex queries
- Multiple display and visualisation modes
  - plain text view
  - XML view
  - graphical tree view
  - time-line view
- Ajax (Asynchronous JavaScript and XML)
- Query and visualisation extensible through modules
Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers
Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers
Graphical Tree Visualisation

Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers
AnnoLab Multi-layer Query Example

- Lexical layer - find the verb *will* ('V')
- Field layer - find Vorfelds ('VF')
- Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

```
let $verb := ds:layer('Lexical')//tok
    [starts-with(pos/text, 'V')]
    [.//orth = 'will']

let $vf := ds:layer('Field')//ntNode
    [category='VF']

return seq:containing($vf, $verb)
```

TUEBA1: Find the verb *will* in the Vorfeld
AnnoLab Multi-layer Query Example

- Lexical layer - find the verb *will* ('V')
- Field layer - find Vorfelds ('VF')
- Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

```
let $verb := ds:layer('Lexical')//tok
     [starts-with(pos/text,'V')]
     [./orth = 'will']

let $vf := ds:layer('Field')//ntNode
     [category='VF']

return seq:containing($vf, $verb)
```

TUEBA2: Find the verb *will* in the Vorfeld
Lexical layer - find the verb *will* ('V')

Field layer - find Vorfelds ('VF')

Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

```
let $verb := ds:layer('Lexical')//tok
    [pos/text = oc:expand('Verb')]
    [.//orth = 'will']

let $vf := ds:layer('Field')//ntNode
    [category='VF']

return seq:containing($vf, $verb)
```

TUEBA2: Find the verb *will* in the Vorfeld using OLiA
corpus query ... oc:expand('Noun') ...

ontology lookup:
1. instance retrieval
2. application of set operators

... NOM_inan | NOM_inan_lq | NOM_anim
| NOM_anim_lq | NOM_anim_pers | NOM_pers | NAME ...

Ontology-Based XQuery’ing of XML-Encoded Language Resources on Multiple Annotation Layers
Experimentation queries

- PQ1 – Get all sentences that contain the word *kam*
- PQ2 – Get all sentences that do not contain *kam*
- PQ3 – Get references to all NPs
- PQ4 – Get all subtrees dominated by NPs
- PQ5 – Get all NPs subtrees dominated by a VP
- TUEBA1 – Find all occurrences of the verb *will* in the Vorfeld
- TUEBA2 – TUEBA1 using OLiA
- BQ2 – Get NPs that are immediate following siblings of a verb
Times normalised to 1000 tokens
PQ1 and PQ2 do not scale
Tested on TüBa-D/Z treebank
PQ1 and PQ2 do not scale

PQ1 – Get all sentences that contain the word *kam*

PQ2 – Get all sentences that do not contain *kam*
Summary

- Approach to querying XML-annotated corpora using standard techniques such as XPath and XQuery
- Extended an XML database to query multi-rooted trees
- Built an OWL ontology of linguistic annotations generalising over annotation schemes and tag sets
- OWL ontology can be used for query expansion
- Implemented an intuitive and flexible graphical query interface
Conclusions and Future Work

- Work on SPLICR is ongoing
- Building the GUI to explore and to query meta-data
- Extended query interface functionality (e.g. saved searches)
- Working on benchmark queries for evaluating XML databases with respect to linguistic corpora