Annotation Interoperability III
Hands-on session

Christian Chiarcos
chiarcos@informatik.uni-frankfurt.de

EUROLAN-2015, 2015, July 22, Sibiu, Romania
Annotation Interoperability

Tue, July 21\textsuperscript{st}, 09:00-10:30
  Annotation Interoperability I
  Ontologies of Linguistic Annotation – Motivations and Principles

Wed, July 22\textsuperscript{nd}, 11:00-12:30
  Annotation Interoperability II
  Applications and Use Cases of Linked Data in Linguistics

Wed, July 22\textsuperscript{nd}, 14:00-15:30
  Annotation Interoperability III
  Hands-on session: Building and Linking OLiA Annotation Models
Idea

• develop OLiA Annotation Models using Protégé
• link them manually to the OLiA Reference Model
• on parts-of-speech and morphosyntactic features from (language-specific) editions of the “Universal Dependencies”
  – http://universaldependencies.github.io/docs/
Prerequisites

• Protégé 5.0 (Desktop version)
  http://protege.stanford.edu/

• text editor supporting Un*x-style line breaks
  – Windows users: not Notepad, better Notepad++
    • https://notepad-plus-plus.org/

• internet connection
1. Building an OLiA Annotation Model for parts of speech (from scratch)
2. Building an OLiA Annotation Model for morphosyntactactic features (from scratch)
3. Exploring the OLiA Reference Model
4. Linking an existing Annotation Model
Building an Annotation Model from Scratch
– Parts of Speech –

http://universaldependencies.github.io/docs/u/pos/all.html

• Open Protégé
  – new ontology
  – add some meta data
  – import ontology http://purl.org/olia/system.owl with prefix olia_system

• Create top-level concept „POS“
  – Create lower levels of hierarchy following document/tagset structure
  – Preserve the original definition as rdfs:comment

• Create an individual for a POS tag
  – olia_system:hasTag
  – olia_system:hasTier = „POS“
Building an Annotation Model from Scratch – Morphosyntactic Features –

http://universaldependencies.github.io/docs/u/feat/all.html

• Create top-level concept „Feat“
  – Create subconcepts concepts for individual features following document/tagset structure
  – Preserve the original definition as rdfs:comment

• For every Feat category $X$ you have, add a has$X$ Object property
  – define domain and range

• Create an individual for a feature value
  – olia_system:hasTag
  – olia_system:hasTier

• don’t forget to save!
Exploring the OLiA Reference Model

• browse/search [http://purl.org/olia/olia.owl](http://purl.org/olia/olia.owl) with Protégé
  – pick a concept you just dealt with

• don’t find it?
  – use your text editor (on a local copy)
Linking an existing Annotation Model

check out „My Documents“ / „EUROLAN 2015“

Limited interoperability between annotations across different annotation schemes both within as well as across languages represents a major hurdle in the development of truly interoperable NLP tools and interoperable corpus querying. I will discuss the problem, motivate and describe the Ontologies of Linguistic Annotation (OLIA) as a component of a possible solution in the context of the Linguistic Linked Open Data (LLOD) cloud and sketch innovative, ontology-based approaches for NLP tasks such as cross-tagset morphosyntactic analysis. The OLIA ontologies represent a repository of annotation terminology for various linguistic phenomena on a great band-width of languages, they have been used to facilitate interoperability and information integration of linguistic annotations in corpora, NLP pipelines, and lexical-semantic resources. In the accompanying hands-on session, participants will be trained to implement new OLIA Annotation Models and their linking to the OLiA Reference Model.
Pick your preferred language

- several language-specific ttl files
- save pos and feat ttl files
- create a new ontology importing both and http://purl.org/olia/olia.owl
Manual Linking

• Go through every concept from the Annotation Models
  – find an appropriate super category
  – assign it a super class (simple or complex)
  – if not possible, leave a comment

• Every concept annotated in the newly created Linking Model has marked in **bold** in ClassView
  – iterate until all has been processed

• Same for Object Properties
  – assign superproperties