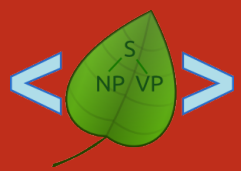
CLARIAH Interoperability: FoLiA



CLST | Centre for Language and Speech Technology
Radboud University

Introduction

Introduction

FoLiA is a rich XML-based format for linguistic annotation

Intended Applications

- as a corpus **storage** format
- as a language resource **exchange** format

Aim

- Provide one solution for a wide variety of linguistic annotation types, using a generic paradigm
- Provide one canonical and validatable serialisation (XML-based)
- Provide a **practical** solution that can be readily adopted
  - FoLiA is developed bottom-up alongside tools and libraries and extended as needed

Characteristics (1/3)

FoLiA Paradigm

Consistent and **generic** framework accross various linguistic annotation types.

- **Seperation of label semantics** from the format itself. FoLiA does not commit to any label set or language. Label sets are defined in separate *set definitions*. Annotations carry *classes* that are defined by these sets.
- **Specific** – Various specific annotation types are defined.
- **Expressive** – Verbose expression of annotations: globally unique identifiers, their annotators, timestamps, confidence values, etc... Support for *alternative* annotations, corrections of any annotation.
- **Formalised** – Validation on shallow (structural) or deep level. The latter validates the label set and allows for links with for data category registries.

Characteristics (2/3)

FoLiA Format

- **Document-based** – One document, one XML file
- Retains **structural** information (like TEI, but not as expressive), **integrated** approach.
- **Localised** – Annotations are stored in close proximity to the textual items they describe. This facilitates streaming parsers.

Characteristics (3/3)

Annotation types

Annotation types can be grouped as follows:

- **Structural Annotation** – Annotation of document structure: divisions, paragraphs, sentences, words/tokens, lists, tables, etc..
- **Token Anotation** – Annotation pertaining to a token (or other structural element): part-of-speech, lemmas, lexical semantic senses
- **Span Annotation** – Annotation spanning multiple tokens: (named) entities, dependency relations, semantic roles, coreference relations, syntactic parsers
- **Higher-order annotations** – Annotations on annotations: corrections, alternatives, comments, descriptions, references to internal or external sources (alignments), annotation of arbitrary substrings

Context: Metadata

Metadata

Documents can ...

- ...refer to external metadata files (any format)
- ...embed other XML-based metadata schemes (e.g. CMDI, Dublin Core)
- ...use FoLiA's native metadata scheme (a simple key value store)

Context: Set Definitions

Set definitions

- Documents declare what annotation types they make use of and what sets are used.
- ...declarations refer to external set definition files
- A set definition defines what classes are part of a particular label set.
- ... classes may be linked to data category registries/ontologies for formal semantic closure.
- Annotations refer to classes
- Set definitions are currently expressed in a simple XML format
- Considering moving these to RDF

Context: Converters (1/2)

Converters: To FoLiA from ...

- Alpino XML
- ALTO XML (OCR output)
- DCOI (legacy)
- HTML (via INL's *OpenConvert*, limited)
- HOCR (HTML OCR output from *Tesseract*)
- NAF (work-in-progress)
- plain text (via tokeniser *ucto* or *OpenConvert* (INL))
- Political Mashup XML
- ReStructuredText
- TEI (via INL's *OpenConvert*, limited)
- MS Word (via INL's *OpenConvert*, limited)

Context: Converters (2/2)

Converters: From FoLiA to ...

- DCOI
- HTML (for rich visualization)
- NAF (work-in-progress, by Antske)
- plain text
- plain text enriched with token annotations (limited)
- simple columned format (one token per line, limited)
- ReStructuredText
- TEI (via INL's *OpenConvert*, limited)

Context: Software Libraries

Software Libraries

- for Python: part of *PyNLPl*
- for C++: *libfolia* (by Ko van der Sloot, RU)

Context: Utilities

Utility collections

- foliatools (python based)
- foliautils (c++ based)

Utilities (CLI)

- **foliavalidator** or **folialint** - FoLiA validators
- foliacorrect (tool to deal with corrections in FoLiA)
- foliacount (compute simple statistics)
- folia-idf (IDF statistics for a directory of FoLiA documents)
- folia-langcat (language detection)
- foliamerge (merge multiple folia documents)
- foliaquery (query FoLiA documents, using FQL or CQL)
- foliacorrect (tool to deal with corrections in FoLiA)
- various of the earlier mentioned converters

Context: FoLiA-aware Software (1/3)

FoLiA-aware NLP Software

- **ucto** - Tokeniser, multilingual, can read and produce FoLiA
- **Frog** - NLP Suite for dutch: tokenisation (via ucto), PoS-tagging, lemmatisation, morphological analysis, shallow parsing, named entity recognition, dependency parsing. Can read and output FoLiA.
- **Gecco** - Generic Environment for Context-aware Correction of Orthography: spelling correction software, backend for **Valkuil.net**. Outputs FoLiA.
- **TICCL** - Text-induced Corpus Cleanup: Text normalisation and post-OCR correction system. Outputs FoLiA.
- **Colibri Core** - Pattern counting and extraction, limited FoLiA input support

Context: FoLiA-aware Software (2/3)

Search software

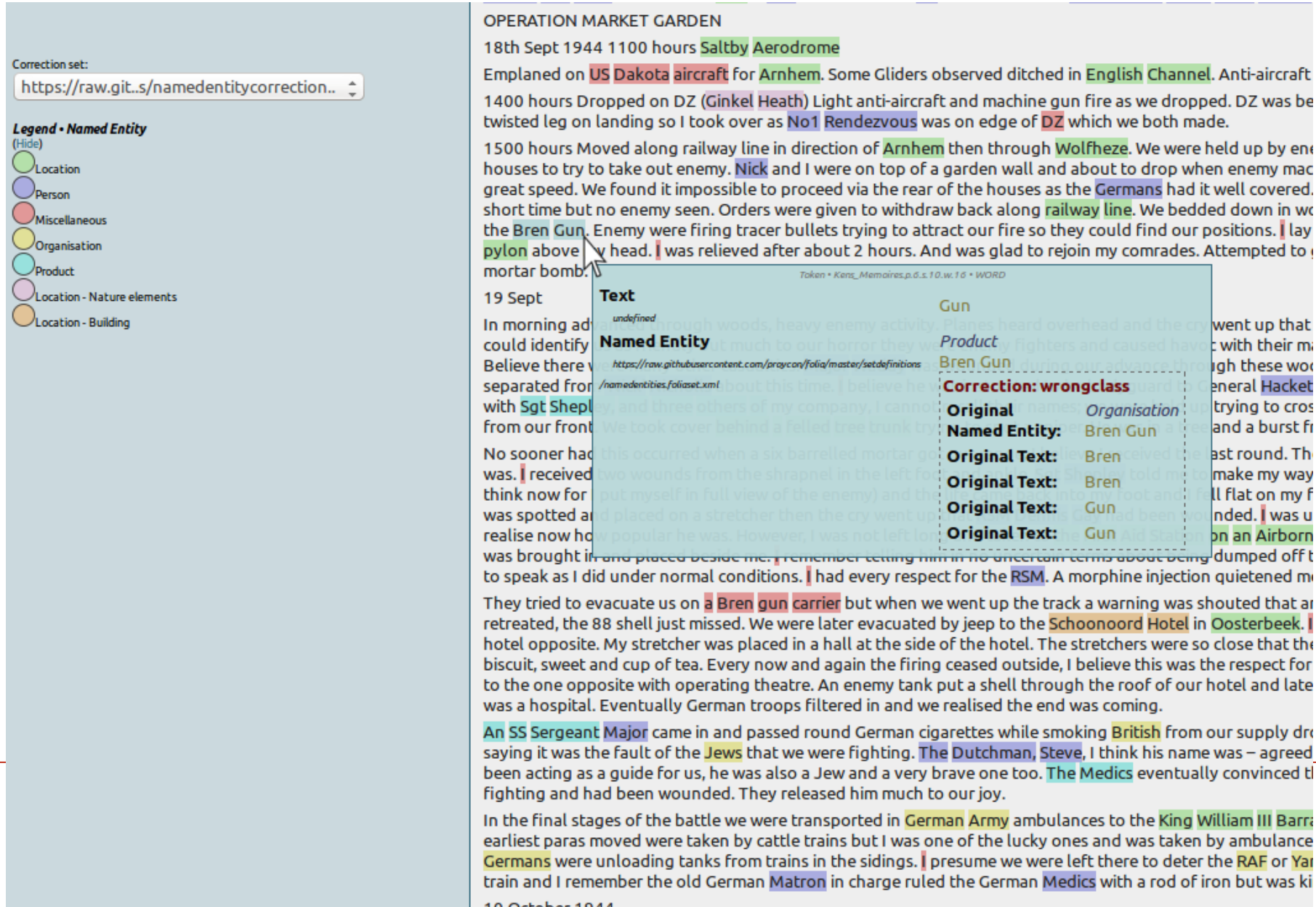
- **BlackLab** - Corpus retrieval engine with (limited) indexing support for FoLiA (INL)
- **WhiteLab** - Web application for the search and exploration of large corpora with linguistic annotations (Matje van de Camp, TaalMonsters)
- **PaQu** - Parse and search in syntactically annotated corpora (Groningen University)

Context: Folia-aware Software (3/3)

FLAT (FoLiA Linguistic Annotation Tool)

- A web-based collaborative annotation tool
- Eventually aims to support annotation for all annotation types supported in FoLiA.
- Used for multiple projects; recently adopted by PARSEME for annotation of MWEs as well
- backend: FoLiA Document Server (*foliadocserve*)
- communication: **FoLiA Query Language (FQL)**
- Public installation: `http://flat.science.ru.nl`

Context: Folia-aware Software (3/3)



Discussion: Interoperability

Aims for the future

- Continued work to extend the format and libraries on a need-to basis
- Continued support to all parties using FoLiA
- Continued work on FLAT
- Finish the FoLiA/NAF conversion tools
- Migrate set definitions to RDF?
- What is the desireability of more RDF support? I.e. RDF schema for FoLiA and conversion to RDF for annotated documents. Use cases?

Discussion: Questions?