FoLiA: Format for Linguistic Annotation

Maarten van Gompel
Radboud University Nijmegen

20-01-2012

Introduction

What is FoLiA?

- Generalised XML-based format for a wide variety of linguistic annotation

Characteristics

- **Generalised** paradigm – Single universal paradigm applicable to all kinds of annotations; as few ad-hoc provisions as possible. Not commited to any label set.
- **Extensible** – Unsupported annotation types can be added fairly easily.
- **Expressive** – Verbose expression of annotations, their annotators, timestamps, etc... Moreover:support for *alternative* annotations.
- **Formalised** – Validation on two levels: shallow and deep. The latter validates the used label set and allows for links with for instance ISOcat.

Properties

- One document, one text, one XML file – containing all annotations.
- Annotation types and label sets must be declared in the document header.
- Document metadata can be either included in the file (limited), or by reference to external CMDI or IMDI (preferred)

Applications

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

Trade-off: Expressibility versus Computing Efficiency

- FoLiA aims at expressibility rather than computing efficiency.
    - XML and FoLiA overhead: Not ideal for real-time or resource-constrained applications
    - **Conversion** to less expressive, more efficient, formats.

Why (yet) another format?

- Many ad-hoc and legacy annotation formats (CGN, Tadpole column format)
- Many theoretic and specialised annotation formats with limited scope (LAF, SynAF, MAF, TEI)
- Bottom-up rather than top-down development: FoLiA arose from practical need, immediately developed alongside practicl programming libraries and applications.
- De-facto-standard: D-COI XML

Dissemination

- SoNaR
- TTNWW
- DutchSemCor
- Valkuil.net
- Frog & Ucto

Supported Annotations (1/2)

FoLiA supports the following linguistic annotations:

- Part-of-Speech tags (with features)
- Lemmatisation
- Spelling corrections on both a tokenised as well as an untokenised level.
- Domain tagging
- Lexical semantic sense annotation (used in DutchSemCor)
- Named Entities / Multi-word units (used in SoNaR)

Supported Annotations (2/2)

FoLiA supports the following linguistic annotations:

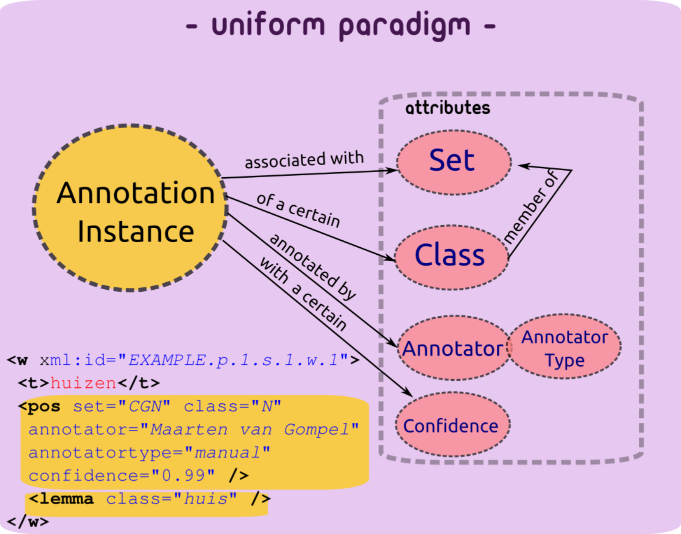
- Syntactic Parses
- Dependency Relations
- Chunking
- Corrections (used in valkuil.net)
- Morphology
- Event annotatation

Paradigm

Paradigm: Annotation Categories

Four categories of annotation:

- **Structure Annotation** - Elements denoting document structure
  - E.g: Divisions, Header, Paragraphs, Sentences, Lists, Figures, Gaps, Quote
- **Token Annotation** - Linguistic Annotations pertaining to a single token (inline annotation)
  - E.g: Part of Speech Annotation, Lemma Annotation, Lexical Semantic Sense Annotation
- **Span Annotation** - Linguistic Annotations spanning over multiple tokens (standoff annotation)
  - E.g: Syntactic Parses, Dependency Relations, Entities/Multi-word Units
- **Subtoken Annotation** - Linguistic Annotations pertaining to a subpart of a token (standoff annotation)
  - E.g: Morphology



```
<w xml:id="EXAMPLE.p.1.s.1.w.1">
 <t>huizen</t>
 <pos set="CGN" class="N"
  annotator="Maarten van Gompel"
  annotatortype="manual"
  confidence="0.99" />
 <lemma class="huis" />
</w>
```

Format

```
<?xml version="1.0" encoding="utf-8"?>
<FoLiA xmlns="http://ilk.uvt.nl/FoLiA"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xml:id="example">
  <metadata type="cmdi" src="example.cmdi">
    <annotations>
       ...
    </annotations>
  </metadata>
  <text xml:id="example.text">
      ...
  </text>
</FoLiA>
```

Example

```
<p xml:id="TEST.p.1">
<t>This is a test. It has two sentences.</t>
<s xml:id="TEST.p.1.s.1">
    <t>This is a test.</t>
    <w xml:id="TEST.p.1.s.1.w.1"><t>This</t></w>
    <w xml:id="TEST.p.1.s.1.w.2"><t>is</t></w>
     ..
</s><s xml:id="TEST.p.1.s.2">...</s>
</p>
```

Characteristics of basic structure

1. **Structure Elements**: Paragraphs, Sentences, Words/Tokens
2. More: Division, Head, List, ListItem, Figure, Gap...
3. Unique identifiers
4. Text content element (t) holds actual text. May occur untokenised on higher levels as well.

Token Annotation

Token annotation occurs within the scope of a word/token (*w*) element.

Example

PoS and Lemma Annotation:

```
<w xml:id="example.p.1.s.1.w.2">
    <t>boot</t>
    <pos set="brown" class="n"
     annotator="Maarten␣van␣Gompel" annotatortype="manual" />
    <lemma set="english-lemmas" class="boot" />
</w>
```

Token Annotations with subsets

For all annotation types; **subsets** can be used for more refined annotations.

Example

```
<w xml:id="example.p.1.s.1.w.2">
    <t>boot</t>
    <pos set="cgn" class="N(soort,ev,basis,zijn,stan)">
     <feat subset="head" class="N" />
        <feat subset="ntype" class="soort" />
        <feat subset="number" class="ev" />
        <feat subset="degree" class="basis" />
        <feat subset="gender" class="zijd" />
        <feat subset="case" class="stan" />
    </pos>
</w>
```

Span Annotation

- Token Annotation not sufficient, some annotations span over multiple tokens
- Spanning multiple tokens can produce nesting problems (e.g $A(BC)D$ and $AB(CD)$)
- **Solution:** Span Annotation using standoff notation
- **Applications:** Syntactic Parses, Chunking, Dependency Relations, Entities/Multi-Word Units
- **Layers:** Each type of span annotation is placed within an *annotation layer*, annotation layers are usually embedded within *sentences* (`s`))
- Same paradigm: Set, class, annotator, confidence, etc...

Span Annotation

```
<s xml:id="example.p.1.s.1">
  <t>The Dalai Lama greeted him.</t>
  <w xml:id="example.p.1.s.1.w.1"><t>The</t></w>
  <w xml:id="example.p.1.s.1.w.2"><t>Dalai</t></w>
  <w xml:id="example.p.1.s.1.w.3"><t>Lama</t></w>
  <w xml:id="example.p.1.s.1.w.4"><t>greeted</t></w>
  <w xml:id="example.p.1.s.1.w.5"><t>him</t></w>
  <w xml:id="example.p.1.s.1.w.6"><t>.</t></w>
  <entities>
    <entity xml:id="example.p.1.s.1.entity.1" class="person">
        <wref xml:id="example.p.1.s.1.w.2" />
        <wref xml:id="example.p.1.s.1.w.3" />
    </entity>
  </entities>
</s>
```

```
<syntax>
<su xml:id="example.p.1.s.1.su.1" class="s">
  <su xml:id="example.p.1.s.1.su.1_1" class="np">
      <su xml:id="example.p.1.s.1.su.1_1_1" class="det">
         <wref xml:id="example.p.1.s.1.w.1" />
      </su>
      <su xml:id="example.p.1.s.1.su.1_1_2" class="pn">
         <wref xml:id="example.p.1.s.1.w.2" />
         <wref xml:id="example.p.1.s.1.w.3" />
      </su>
   </su>
 </su>
 <su xml:id="example.p.1.s.1.su.1_2" class="vp">
    <su xml:id="example.p.1.s.1.su.1_1_1" class="v">
        <wref xml:id="example.p.1.s.1.w.4" />
    </su>
    <su xml:id="example.p.1.s.1.su.1_1_2" class="pron">
      <wref xml:id="example.p.1.s.1.w.5" />
    </su>
 </su>
</su>
</syntax>
```

Tools for working with FoLiA

- Standard **XML** facilities: XSLT, XPath
- **Python** library: pynlpl.formats.folia
- **C++** library: libfolia *(Ko van der Sloot)*

Applications

- **Frog** – tagger/lemmatisaion/parser suite: FoLiA output (input in later stage).
- **ucto** – tokeniser: FoLiA input and output.

Converters

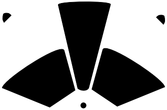
- DCOI ⟷ FoLiA
- FoLiA ⟶ CSV (limited)

Conclusion

- **Uniformity:** generic framework with simple paradigm, XML based
- **Expressiveness:** Ability to encode many kinds of linguistic annotation, including structural annotation, alternatives, and corrections
- **Expandibility:** easy to add new annotations with the same paradigm
- A variety of tools and converters already available!

URLs

- http://ilk.uvt.nl/folia
- http://github.com/proycon/folia



Questions?

Token Annotation

All annotations need to be declared in the metadata:

- Default sets and annotator *may* be predefined at this level

Example

```
<metadata>
 <annotations>
  <token-annotation />
  <pos-annotation set="brown" annotator="Maarten␣van␣Gompel"
    annotatortype="manual"/>
  <lemma-annotation />
 </annotations>
</metadata>
```

Alternative Token Annotations

Annotations of the same type, but different sets need *not* be alternatives.

```
<w xml:id="example.p.1.s.1.w.2">
    <t>luid</t>
    <pos set="brown" class="jj" />
    <pos set="cgn" class="adj" />
</w>
```

There can be only one of the same set though, this is illegal and requires usage of alternatives instead:

```
<w xml:id="example.p.1.s.1.w.2">
    <t>luid</t>
    <pos set="cgn" class="adj" />
    <pos set="cgn" class="adv" />
</w>
```

Alternative Token Annotations

Encodes mutually exclusive alternative annotations. Any annotations that are not alternatives are considered “selected”.

```
<w xml:id="example.p.1.s.1.w.2">
    <t>bank</t>
    <sense set="wordnet3.0" class="bank%1:17:01:"
     annotator="Maarten␣van␣Gompel" annotatortype="manual"
     confidence="0.8">
     sloping ground near water</sense>
    <alt xml:id="example.p.1.s.1.w.2.alt.1">
     <sense set="wordnet3.0" class="bank%1:14:01:"
      annotator="WSDsystem" annotatortype="auto"
      confidence="0.6">
      financial institution</sense>
    </alt>
</w>
```

Alternative Token Annotations

All token annotations grouped as one alternative are considered dependent. Multiple alternatives are always independent:

```
<w xml:id="example.p.1.s.1.w.2">
    <t>vlieg</t>
    <pos class="N" />
    <lemma class="vlieg" />
    <alt xml:id="example.p.1.s.1.w.2.alt.1">
        <pos class="V" />
        <lemma class="vliegen" />
    </alt>
</w>
```