

Corpus Tools: Annotation, tools

Max Ionov
max.ionov@gmail.com

Goethe-Universität Frankfurt

25.03.2017

Table of Contents

1 Corpus annotation

2 Tools overview

Annotation basics

- Annotation allows us to search efficiently

Annotation basics

- Annotation allows us to search efficiently
- Manual or automatic

Annotation basics

- Annotation allows us to search efficiently
- Manual or automatic
- There is no *right way* to annotate

Annotation basics

- Annotation allows us to search efficiently
- Manual or automatic
- There is no *right way* to annotate
- Layers of annotation:
 - Text-level annotation

Annotation basics

- Annotation allows us to search efficiently
- Manual or automatic
- There is no *right way* to annotate
- Layers of annotation:
 - Text-level annotation
 - Segment-level annotation

Text-level annotation

- Annotates text as a whole

Text-level annotation

- Annotates text as a whole
- Metadata:

Text-level annotation

- Annotates text as a whole
- Metadata:
 - Source
 - Name of the text (?)
 - Author
 - Date
 - Genre / Type / Style

Segment-level annotation

- First, we should annotate segments:

Segment-level annotation

- First, we should annotate segments:
 - Words
 - Sentences
 - Paragraphs

Segment-level annotation

- First, we should annotate segments:
 - Words
 - Sentences
 - Paragraphs
 - Verses
 - Discourse elements
 - Questions
 - ...

Segment-level annotation

- Annotation of a segment

Segment-level annotation

- Annotation of a segment
 - Part of speech
 - Morphology
 - Errors
 - ...

Segment-level annotation

- Annotation of a segment
 - Part of speech
 - Morphology
 - Errors
 - ...
- Annotation of a relation between two segments
 - Dependency syntax
 - Anaphora / coreference relations
 - Discourse relations (e.g. RST)

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task
- Example: coordination and dependency syntactic annotation

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task
- Example: coordination and dependency syntactic annotation
 - What should be a head?

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task
- Example: coordination and dependency syntactic annotation
 - What should be a head?
 - What is a parent of a node?

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task
- Example: coordination and dependency syntactic annotation
 - What should be a head?
 - What is a parent of a node?
- There always should be a thorough manual on how to annotate

What is a right way to annotate?

- There is not *right way*
- Depends heavily on a task
- Example: coordination and dependency syntactic annotation
 - What should be a head?
 - What is a parent of a node?
- There always should be a thorough manual on how to annotate
- Especially, if this is a group effort

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN
- An important aspect: where to store

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN
- An important aspect: where to store
 - In-place annotation

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN
- An important aspect: where to store
 - In-place annotation: keeping annotations with a text

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN
- An important aspect: where to store
 - In-place annotation: keeping annotations with a text
 - Stand-off annotation

How to store annotations?

- Usually depends on a tool
- A bunch of standard ways:
 - Low-level: XML, JSON, Text-based, Database-based
 - High-level: CONLL, ANNIS, Toolbox / FLex / ELAN
- An important aspect: where to store
 - In-place annotation: keeping annotations with a text
 - Stand-off annotation: keeping annotations in a different place with offset and length of a segment

Annotation Examples: in-place

- Morphological annotation in LOB Corpus:

```
hospitality_NN is_BEZ an_AT excellent_JJ virtue_NN ,_,  
but_CC
```

Annotation Examples: in-place

- Morphological annotation in LOB Corpus:

hospitality_NN is_BEZ an_AT excellent_JJ virtue_NN ,_,
but_CC

- Morphological annotation in National Russian Corpus:

Я{я=S, ед, од=им}

сидел{сидеть=V, несов=изъяв, прош, ед, муж} на{на=PR}

барском{барский=A=ед, сред, пр}

сиденье{сиденье=S, сред, неод=ед, пр}

Annotation Examples: in-place

- Morphological annotation in LOB Corpus:
hospitality_NN is_BEZ an_AT excellent_JJ virtue_NN ,_,
but_CC
- Morphological annotation in National Russian Corpus:
Я{я=S, ед, од=им}
сидел{сидеть=V, несов=изъяв, прош, ед, муж} на{на=PR}
барском{барский=A=ед, сред, пр}
сиденье{сиденье=S, сред, неод=ед, пр}
- Syntactic annotation + morphology
[S[NP Claudia_NP1 NP] [VP sat_VVD [PP on_II [NP a_AT1
stool_NN1 NP] PP] VP] S]

Annotation Examples: stand-off

Coreference annotation in RuCor corpus:

doc_id	variant	group_id	chain_id	link	shift	length	content	tk_shifts
1	1	407840	1070	0	9	5	своих	9
1	1	407839	1070	407840	47	1	я	47
1	1	407842	1069	0	69	13	одинокую дачу	69,78
1	1	407841	1069	407842	118	9	этой даче	118,123
1	1	407843	1069	407841	166	3	она	166
1	1	407846	1067	0	184	15	высоким забором	184,192
1	1	407844	1068	0	203	15	низкой калиткой	203,216,223
1	1	407845	1068	407844	233	7	которая	233
1	1	407847	1067	407846	316	7	забором	316

in-place vs. stand-off

- in-place cons:

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools
 - Harder to add another layer of annotation

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools
 - Harder to add another layer of annotation
 - Harder to work separately on different layers

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools
 - Harder to add another layer of annotation
 - Harder to work separately on different layers
- stand-off cons:
 - A lot of opportunities to screw everything up

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools
 - Harder to add another layer of annotation
 - Harder to work separately on different layers
- stand-off cons:
 - A lot of opportunities to screw everything up
 - Harder to maintain: a lot of different files

in-place vs. stand-off

- in-place cons:
 - Harder to read the corpus without any tools
 - Harder to add another layer of annotation
 - Harder to work separately on different layers
- stand-off cons:
 - A lot of opportunities to screw everything up
 - Harder to maintain: a lot of different files
 - Harder to see the annotation as a whole (without a tool)

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user
 - Annotation procedure should be known to the end user

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user
 - Annotation procedure should be known to the end user
 - Annotation is an interpretation, not necessarily the truth

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user
 - Annotation procedure should be known to the end user
 - Annotation is an interpretation, not necessarily the truth
 - Annotation should be theory-neutral (in most of the cases)

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user
 - Annotation procedure should be known to the end user
 - Annotation is an interpretation, not necessarily the truth
 - Annotation should be theory-neutral (in most of the cases)
 - Annotation guidelines should be created according to the practical reasons, not just because we want to do it like this

Annotation principles

- Leech's Maxims of Annotation (Leech 1993)
 - Annotation should be separable from the text
 - Annotation guidelines should be known to the end user
 - Annotation procedure should be known to the end user
 - Annotation is an interpretation, not necessarily the truth
 - Annotation should be theory-neutral (in most of the cases)
 - Annotation guidelines should be created according to the practical reasons, not just because we want to do it like this
- Those are only recommendations, in reality some of them are often violated

Table of Contents

1 Corpus annotation

2 Tools overview

2 main types of corpus tools:

2 main types of corpus tools:

- Web-based: ANNIS, Sketchengine, Nosketchengine, *BRAT*

2 main types of corpus tools:

- Web-based: ANNIS, Sketchengine, Nosketchengine, *BRAT*
- Standalone: MMAX, Exmeralda, *UAM Corpus Tool*
- Web-based:
 - harder to set up
 - + allows simultaneous work
 - + easier to share

2 main types of corpus tools:

- Web-based: ANNIS, Sketchengine, Nosketchengine, *BRAT*
- Standalone: MMAX, Exmeralda, *UAM Corpus Tool*
- Web-based:
 - harder to set up
 - + allows simultaneous work
 - + easier to share
- Standalone
 - + easy to set up
 - harder to organize shared work
 - ± can be harder to share