

A practical workshop on automatic morpho-syntactic annotation of large language corpora using the Universal Dependencies framework

18th of April 2024 University of Tartu, Estonia

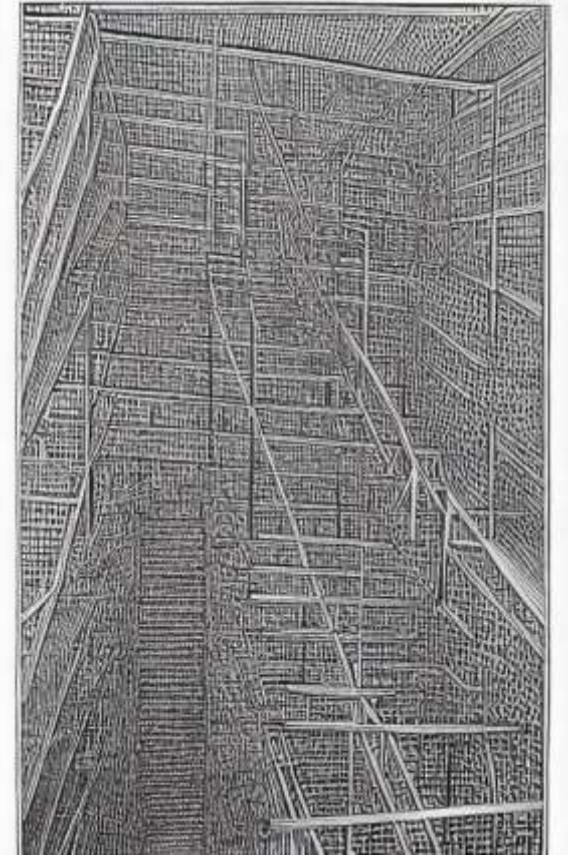
Hanna-Mari Kupari hmknie@utu.fi

➤ Workshop Day 4

➤ Recap of Wednesday

ConlluEditor tool for correcting output

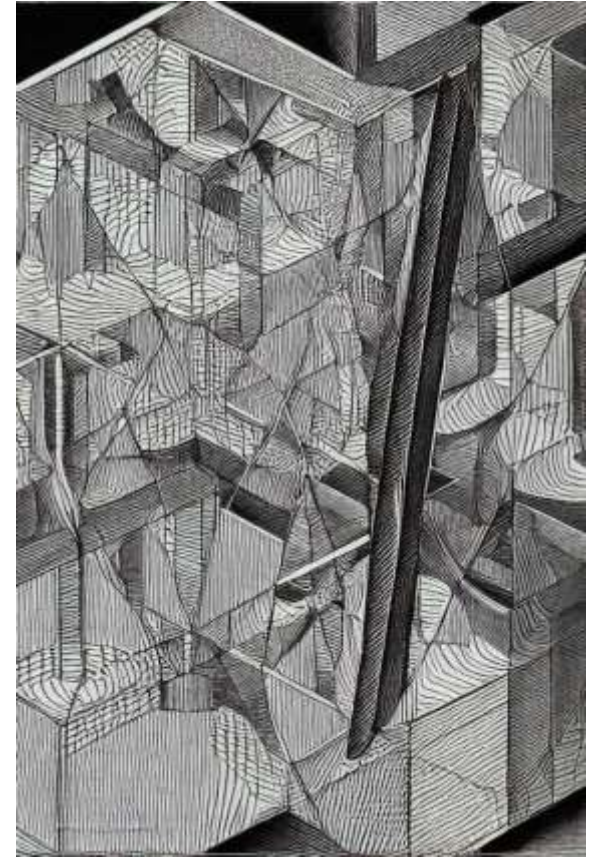
Personal projects help



Deep AI

Shortly about today's aims

- Working in a workshop style – no “theory” today
- Understanding how to implement the UD guidelines for own texts



BRAT

annotation tool

https://brat.nlpab.org/introduction.html


home | introduction | examples | features | manual | site map | contact | brat

mini-introduction to brat

brat is a web-based tool for text annotation; that is, for adding notes to existing text documents.

brat is designed in particular for structured annotation, where the notes are not freeform text but have a fixed form that can be automatically processed and interpreted by a computer.

the following screenshot shows a simple example where a sentence has been annotated to identify mentions of some real-world entities (things) and their types, and a relation between two.




example annotations (following in part the [ACE 2005](#) entity and relation annotation guidelines)

this example illustrates two basic categories of annotation:

- **text span** annotations, such as those marked with the *Organization* and *Person* types in the example
- **relation** annotations, such as the *family* relation in the example

the simple typed text span category is suitable for creating annotations for [named entity recognition](#), and binary relations for simple relational [information extraction](#) tasks, among others.

brat also supports the annotation of **n-ary associations** that can link together any number of other annotations participating in specific roles. This category of annotation can be used for example for event annotation, such as *TRANSFER* in the following example:



Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou and Jun'ichi Tsujii (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*.

Representing language as numbers

- What is a **bag of words** approach?
- Creating a sparse matrix out of vocabulary in corpus
- A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:
 1. A vocabulary of known words.
 2. A measure of the presence of known words.
- <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
- *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Jufarsky 2014

What are embeddings?

- words and documents are represented in the form of numeric vectors allowing similar words to have similar vector representations
- <https://www.turing.com/kb/guide-on-word-embeddings-in-nlp>
- XLM-Roberta Large is the basis of Trankit
- *XLM-RoBERTa* model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages
- <https://huggingface.co/FacebookAI/xlm-roberta-large>

Today's main focus: HOW 3

- For everyone:
- ConlluEditor installation and use
- If you are coming to your first day of workshop:
 - Stanza (working with files)
 - UD pipe 2 on web interface
- Main idea is to understand how to make new Gold Standard datasets

Using Trankit

- https://github.com/HannaKoo/ParsersTartu/blob/main/Trankit_use.md
- <http://nlp.uoregon.edu/trankit> (For beginners)

Using Stanza step by step beginner's level

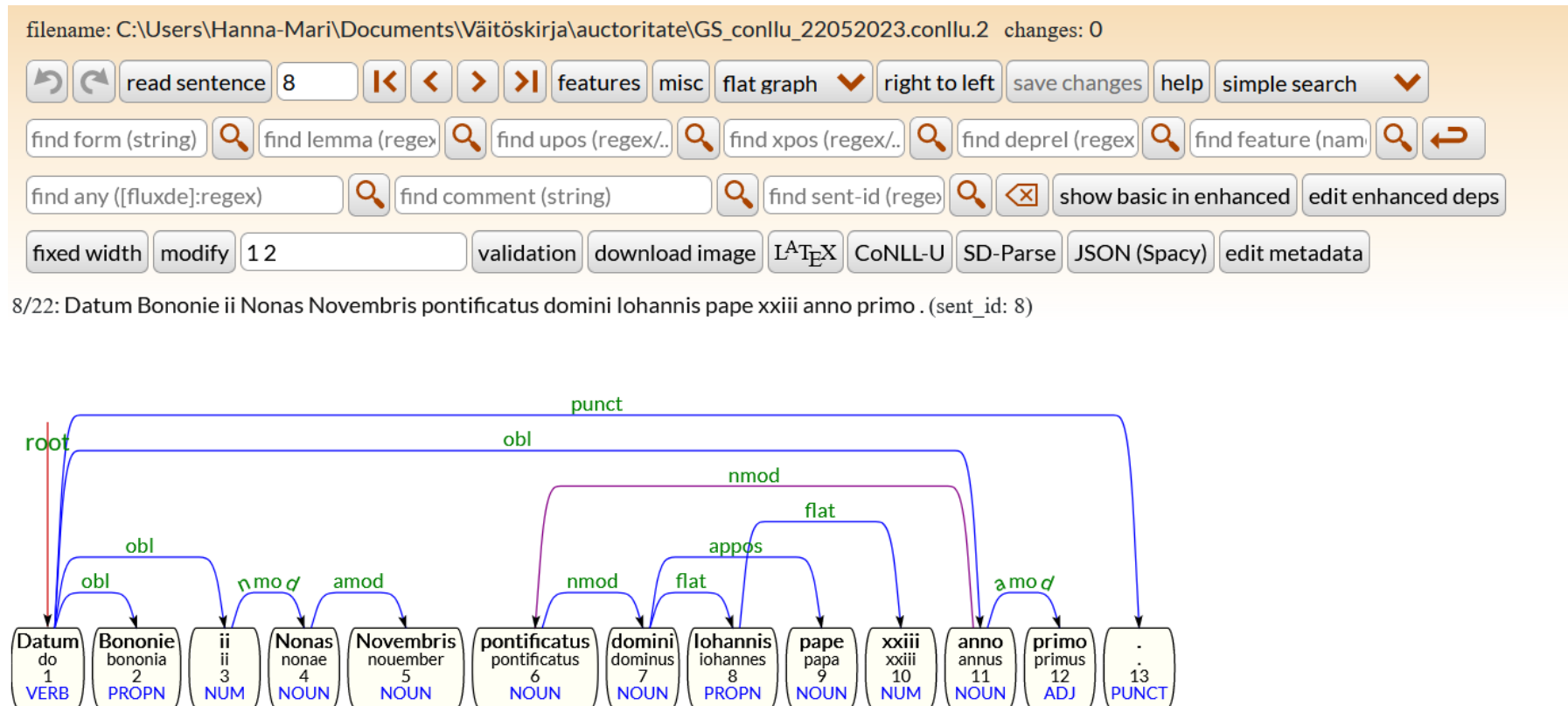
- Have a look at <https://stanfordnlp.github.io/stanza/>
- Open the .md file from https://github.com/HannaKoo/ParsersTartu/blob/main/Stanza_use.md
- We will first work with English and then:
- <https://stanfordnlp.github.io/stanza/performance.html#system-performance-on-ud-treebanks>
- Working with files:
- https://github.com/HannaKoo/ParsersTartu/blob/main/documents_use

Qualitative analysis of predictions

- Making corrections to predictions:
- One perfectly good option is to just look at the CoNLL-U-file
- Making corrections to another column next to the one predicted
- Correcting more complex syntax structures needs a program
- My own personal experience is from ConlluEditor

Today:

- Install Conllu-editor



Detailed instructions to follow

- https://github.com/HannaKoo/ParsersTartu/blob/main/ConlluEditor_use.md
- clicking on a word and then clicking on the head-word creates a dependency relation. An edit window opens to enter the relation a name
- Existing relations can be renamed by clicking on their name
- Clicking twice on a word deletes its eventual dependency relation and makes it root
- <https://github.com/Orange-OpenSource/conllueditor>

Installing ConlluEditor

<https://adoptium.net/temurin/releases/?os=windows&package=jre&version=17&arch=x64>

Eclipse Temurin™ Latest Releases



Eclipse Temurin is the open source Java SE build based upon OpenJDK. Temurin is available for a [wide range of platforms](#) and Java SE versions. The latest releases recommended for use in production are listed below, and are regularly [updated and supported](#) by the Adoptium community. Migration help, container images and package installation guides are available in the [documentation section](#).

Use the drop-down boxes below to filter the list of current releases.

Operating System	Architecture	Package Type	Version
Windows ▾	x64 ▾	JRE ▾	17 - LTS ▾

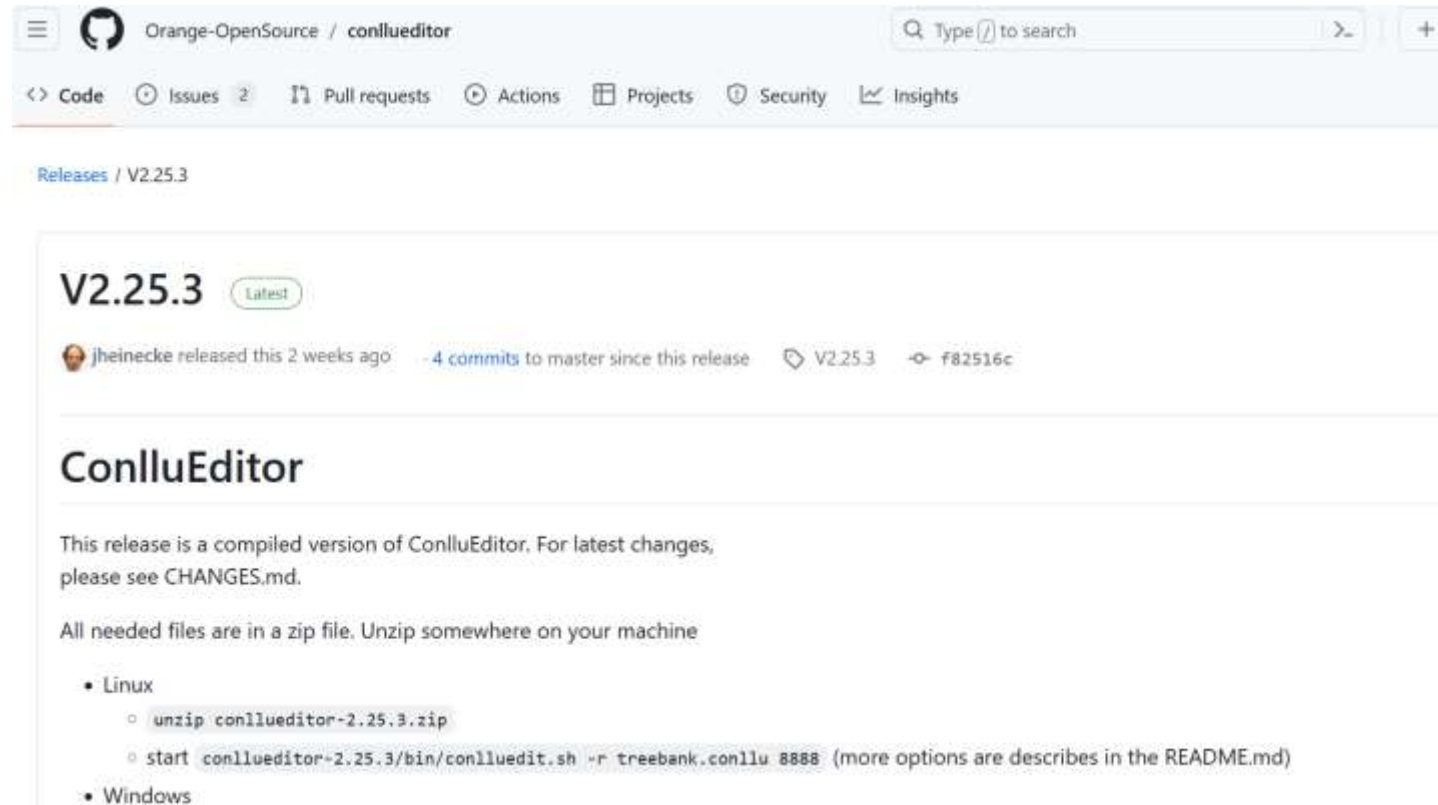
<div>17.0.10+7</div> <div>Temurin </div> <div>January 18, 2024</div>	Windows	x64	JRE - 31 MB Checksum <div> .msi</div>
			JRE - 43 MB Checksum <div> .zip</div>

installation
package

Previous releases are available in the Temurin archive.

Getting the latest release of ConlluEditor

<https://github.com/Orange-OpenSource/conlleditor/releases/tag/V2.25.3>



The screenshot shows the GitHub interface for the 'conlleditor' repository by 'Orange-OpenSource'. The top navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects, Security, and Insights. The main content area displays the release 'V2.25.3' as the 'Latest' version, released by 'jheinecke' 2 weeks ago. It notes '4 commits' to master since this release and provides a download link for 'f82516c'. Below the release information, the title 'ConlluEditor' is followed by a description: 'This release is a compiled version of ConlluEditor. For latest changes, please see CHANGES.md.' and 'All needed files are in a zip file. Unzip somewhere on your machine'. Installation instructions are provided for Linux and Windows.

Orange-OpenSource / conlleditor

Releases / V2.25.3

V2.25.3 Latest

jheinecke released this 2 weeks ago · 4 commits to master since this release · V2.25.3 · f82516c

ConlluEditor

This release is a compiled version of ConlluEditor. For latest changes, please see CHANGES.md.

All needed files are in a zip file. Unzip somewhere on your machine

- Linux
 - `unzip conlleditor-2.25.3.zip`
 - `start conlleditor-2.25.3/bin/conlledit.sh -r treebank.conllu 8888` (more options are describes in the README.md)
- Windows




- start `conllueditor-2.25.3/bin/conlluedit.sh -r treebank.conllu 8888` (more options are describes in the README.md)
- Windows
 - `unzip conllueditor-2.25.3.zip`
 - start server `java -jar conllueditor-2.25.3\target\ConlluEditor-2.25.3-jar-with-dependencies.jar --rootdir conllueditor-2.25.3\gui treebank.conllu 8888`

For windows, you possibly need to add python3 to the validation script

```
script: python3 C:\<path>\validate.py --lang cy --max-err 0 --level 5 {FILE}
```

📄

▼ Assets 3

 conllueditor-2.25.3.zip	11.9 MB	2 weeks ago
 Source code (zip)		2 weeks ago
 Source code (tar.gz)		2 weeks ago





Kieli- ja käännöstieteiden laitos 2024