cost
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

Funded by
the European Union

UniDive Training School Course 3.5

Corpus Quality

Daniel Zeman, Bruno Guillaume

zeman@ufal.mff.cuni.cz
https://unidive.lisn.upsaclay.fr/

Outline

1. Release early, release often
2. Fixing validation errors in text-editors
3. Error mining and correcting with Udapi
4. Error mining and correcting with Grew-match (Bruno; separate slides)

Release Early, Release Often

- Wait until treebank is complete, then ask for repository???
  - Not required
  - Not recommended (though possible)
  - The sooner you familiarize yourself with UD infrastructure, the better
  - If you are ready to annotate your first twenty sentences, it is time to ask for a repository
  - You can always expand it later

Release Early, Release Often

- Wait until treebank is complete, then ask for repository???
  - Not required
  - Not recommended (though possible)
  - The sooner you familiarize yourself with UD infrastructure, the better
  - If you are ready to annotate your first twenty sentences, it is time to ask for a repository
  - You can always expand it later
- It is normal that treebank grow and improve between releases
  - Your first annotations are released
  - You get visibility and possibly feedback
  - There will be another release in six months, so you can fix stuff if necessary!

Fixing Validation Errors

- `https://shorturl.at/8A3T9`
  - (`https://quest.ms.mff.cuni.cz/udvalidator/cgi-bin/unidep/validation-report.pl`)
- Some annotation errors must be fixed in documentation rather than in data!
  - ⇒ Tomorrow I will say how
  - Typically types *feat-*, unknown-deprel, aux-lemma, cop-lemma*
  - In particular at the beginning there may be thousands of them

Fixing Solitary Errors

- Often OK in (plain) text editor
  - CTRL+G may take you to the line number indicated in the error

Fixing Solitary Errors

- Often OK in (plain) text editor
  - CTRL+G may take you to the line number indicated in the error
- Not in Microsoft Word, for example!
  - I use Notepad2 on Windows (`https://www.flos-freeware.ch/notepad2.html`)

Fixing Solitary Errors

- Often OK in (plain) text editor
  - CTRL+G may take you to the line number indicated in the error
- Not in Microsoft Word, for example!
  - I use Notepad2 on Windows (`https://www.flos-freeware.ch/notepad2.html`)
- Danger: Even some plain text editors will auto-convert TABs to spaces (it can be configured)
  - Recall: TABs delimit columns in CoNLL-U. Spaces don't
  - If every TAB is saved as a sequence of spaces, you have 1 column even if it looks like 10

Fixing Repetitive Errors

- Use a tree-processing program (script)
- Instead of writing whole script, use a toolkit such as:
  - UDAPI (next slide)
  - Grew-Rewrite (Bruno's part)

- **Always check `git diff` before committing!**

UDAPI

- https://udapi.github.io/
- Udapi-Python is a Python module that gives you easy access to CoNLL-U files

- Tutorial:
  - https://shorturl.at/VVvZr
  - (https://github.com/UniDive/2024-UniDive-Chisinau-training-school/blob/main/Course-3-corpus-annotation-infrastructure/udapi-tutorial-dz.pdf)

UDAPI

- https://udapi.github.io/
- Udapi-Python is a Python module that gives you easy access to CoNLL-U files
- If you know Python, it will be very easy for you to use
- Tutorial:
  - https://shorturl.at/VVvZr
  - (https://github.com/UniDive/2024-UniDive-Chisinau-training-school/blob/main/Course-3-corpus-annotation-infrastructure/udapi-tutorial-dz.pdf)

UDAPI

- https://udapi.github.io/
- Udapi-Python is a Python module that gives you easy access to CoNLL-U files

- If you know Python, it will be very easy for you to use
- Even if you don't know Python, the few things you need to know are simple

- Tutorial:
  - https://shorturl.at/VVvZr
  - (https://github.com/UniDive/2024-UniDive-Chisinau-training-school/blob/main/Course-3-corpus-annotation-infrastructure/udapi-tutorial-dz.pdf)

Udapi Blocks

- Reading block (default: `read.Conllu`)
- Optional processing blocks (modify the trees, filter the trees)
  - Udapi comes with many useful blocks
    - ★ E.g. `ud.FixPunct`
  - You can add your own
- Optional writing block
  - Default `write.Conllu` if you modified the data and use the `-s` option to save it
  - Possible fancy rendering, such as `write.TextModeTrees`

Udapi Blocks

- Reading block (default: `read.Conllu`)
- Optional processing blocks (modify the trees, filter the trees)
  - Udapi comes with many useful blocks
    - E.g. `ud.FixPunct`
  - You can add your own
- Optional writing block
  - Default `write.Conllu` if you modified the data and use the `-s` option to save it
  - Possible fancy rendering, such as `write.TextModeTrees`

- Blocks may have parameters

Udapi Blocks

- Reading block (default: `read.Conllu`)
- Optional processing blocks (modify the trees, filter the trees)
  - Udapi comes with many useful blocks
    - ★ E.g. `ud.FixPunct`
  - You can add your own
- Optional writing block
  - Default `write.Conllu` if you modified the data and use the `-s` option to save it
  - Possible fancy rendering, such as `write.TextModeTrees`

- Blocks may have parameters

- Instead of modifying the data, the middle blocks can search for something and directly print it