

# GraphAnno: A lightweight tool for multi-level annotation

Volker Gast

February 10, 2017

## Downloading and starting the programme

- ▶ GraphAnno is available on GitHub, maintained by Lennart Bierkandt (LBierkandt/graph-annoo)
- ▶ For today, we will get it from my account, as I have included some demo data for illustration; go to github and search for **grno**
- ▶ Download ZIP
- ▶ Unpack zip-file somewhere
- ▶ Open the folder `grno-master` and start `main.exe`
- ▶ Open browser and go to URL `localhost:4567` (you can add a bookmark if you like)

# Operating GraphAnno

- ▶ There is a comprehensive **documentation** (link in the bottom-right corner)
- ▶ GraphAnno is (basically) operated with the keyboard:
  - ▶ Commands in the command line in the left-bottom corner
  - ▶ Function keys

# Importing data

- ▶ Data can be created manually (but we would not normally do that), e.g. with

```
ns 1 2 3 4 5
```

(create 5 new sentences with name '1, 2, 3, 4, 5'; cf. F9)

```
t This is an example (tokenize "This is an example")
```

- ▶ Import of raw text

```
import text
```

- ▶ Import through GraphPynt (Python interface to GraphAnno; allows preprocessing, e.g. parsing, assignment of tokens to speakers, etc.): Available on GitHub (account VolkerGast)

- ▶ GraphAnno-files are loaded with

```
load $file
```

# Operating GraphAnno

- ▶ Configuration:
  - ▶ `pref`: Customizing autocomplete options
  - ▶ `config`: customize layers
  - ▶ `metadata`: keys and values for metadata-annotation
  - ▶ `tagset`: define tagsets
  - ▶ `makros`: define annotation makros
  - ▶ `annotators`: create annotators (for multi-annotator projects)
  - ▶ `file`: some file settings

# Navigating in GraphAnno

- ▶ Navigating with F-keys:
  - ▶ F6: Filter: allows us to show only specific annotation layers
  - ▶ F7: Search window: Allows us to search for nodes of a specific type and graph fragments
  - ▶ F8: Command history: Records all the commands of a session
  - ▶ F9: Navigation window (shows all the sentences)
- ▶ Navigating with keyboard shortcuts
  - ▶ Navigation within the window is done with CTRL+SHIFT+[X], with [X] standing for
    - ▶ arrow keys: up, down, left, right
    - ▶ +, -: zoom in/out
- ▶ Navigating within the corpus can be done with ALT+[ARROW]

## An example: Spoken language

- ▶ `load demo/11c.json`
- ▶ Toggle navigation window with F9
- ▶ Select different portions of the corpus (e.g. sentences 7 and 8) by selecting them in the navigation window (multiple selection with CTRL [individual] or SHIFT [sequences of sentences]); confirm selection with RETURN.
- ▶ Inspect the data (toggle node identifiers with F4)

## Some basic annotations: Intonation

- ▶ Annotate Intonation Phrases, e.g. for sentence 4  
g n4..n6 cat:IP t  
g: 'group' (create constituent)  
n4..n6 node identifiers, equivalent to n4 n5 n6  
cat:IP category attribute is privileged, as shown without cat-key  
t annotate at 't-layer' (cf. config)
- ▶ We can create an annotation makro, e.g. x for the annotation  
cat:IP  
g n4..n6 x



# Structural annotations

- ▶ Select a different layer, e.g. *s*
- ▶ It might make sense to make intonational annotations invisible with F6
- ▶ Annotate structural units such as the VALESCO-categories *subact*, *act*, *intervention*, e.g.

`g n8..n13 cat:SUBACT s g n5..n7 n31 cat:ACT`

## An example: hierarchical annotations

- ▶ load demo/cc: load file cc.json (with some data on concessives)
- ▶ Toggle internal windows: F6, F7, F8, F9
- ▶ Toggle node identifiers: F4
- ▶ Try to navigate in the window, inspecting the first sentence of the corpus (*Of course they have been put forward by political groups, ...*).
- ▶ Modify the tree with commands such as:
  - ▶ a \$node \$key:\$val\$
  - ▶ di \$node\$ (delete node)
  - ▶ g \$node1 \$node2 \$key:\$val
  - ▶ ni \$node \$edge \$key:\$val

# A final note

- ▶ There is a multi-media version of GraphAnno, fully functional on Linux and (very soon) also available for Windows and Mac!

The screenshot displays the GraphAnno multi-media interface. It includes a list of annotations on the left, a search panel with buttons for 'Search', 'Annotate', 'Clear search', 'Export data table', and 'Export subcorpus', a video frame showing three people, and a detailed syntactic tree for the sentence 'Bisato Nammi bi gta da da ga anyikanyik ga [gpa] da nanyargaa gta adada rajal palai da da [la kn-] la nyi kwankwaja nanki nanyama gta gta ga bo dani nanimarala dani tukama dmen olalo ok bo ade gilon ga kumbabi la gta nyi kwankwaja da nany- nanimarala ok gta la gta palai da riyon ola ga bo plogplog riyonora gta palai da ga tukama ola da da man tirma bo ga ada thona gta gaggasurindin ga ga dani ryon ola oom da panti ara oom ada ara da gta thona yanni gaggasurindin bo ga gaggasurindin gta gta palai da bo ryon ola bo gta nyi kwankwaja thuyabi nanki nanyam ok gta dani waddan'.

The syntactic tree shows the following structure:

- S (Sentence) branches into NP (Noun Phrase) and V (Verb).
- NP branches into PN (Proper Noun), PRO (Pronoun), PTCL (Particle), DEM (Demonstrative), COP (Copula), N (Noun), CASE (Case), PTCL (Particle), NC (Noun Class), COP (Copula), V (Verb), DEM (Demonstrative), A (Adjective), N (Noun), N (Noun), COP (Copula), COP (Copula), NC (Noun Class), NC (Noun Class), N (Noun), N (Noun), NP (Noun Phrase), and CASE (Case).
- PN: Nammi
- PRO: bi
- PTCL: ga
- DEM: gta
- COP: da
- COP: da
- PTCL: ga
- N: anyikanyik
- CASE: ga
- PTCL: [gpa]
- NC: da
- COP: nanyargaa
- V: gta
- DEM: adada
- A: gnd
- N: palai
- N: da
- COP: da
- COP: da
- NC: [la
- NC: kn-]
- N: la
- N: nyi
- NP: kwankwaja
- CASE: n

Below the tree, a table of numerical values is provided for each node:

5.7	6.0	6.4	6.7	7.1	8.4	8.7	9.1	9.4	9.8	10.1	10.5	12.4	12.8	13.1	13.5	13.9	14.2	14.9	15.3	15.6	16.0	16.3	16.7
6.0	6.3	6.7	7.0	7.4	8.7	9.0	9.4	9.7	10.1	10.4	10.8	12.8	13.2	13.5	13.9	14.3	14.6	15.3	15.7	16.0	16.4	16.7	17.1

At the bottom, there is a text area with the sentence: 'Nammi bi gta da da ga anyikanyik ga [gpa] da nanyargaa gta adada rajal palai da da [la kn-] la nyi kwankwaja nanki nanyama gta gta ga bo dani nanimarala dani tukama dmen olalo ok bo ade gilon ga kumbabi la gta nyi kwankwaja da nany- nanimarala ok gta la gta palai da ryon ola ga bo plogplog riyonora gta palai da ga tukama ola da da man tirma bo ga ada thona gta gaggasurindin ga ga dani ryon ola oom da panti ara oom ada ara da gta thona yanni gaggasurindin bo ga gaggasurindin gta gta palai da bo ryon ola bo gta nyi kwankwaja thuyabi nanki nanyam ok gta dani waddan'.

Additional interface elements include a 'name: 3' field, a 'functional layer' dropdown, a 'file: datapicture\_taskpicture\_task\_const.json' field, a 'Search' button, and a 'Documentation' link.

## Some references

- ▶ Gast, V., L. Bierkandt, S. Druskat & C. Rzymiski (2016). 'Enriching TimeBank: Towards a more precise annotation of temporal relations in a text'. In Nicoletta Calzolari et al. (eds.): *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association.
- ▶ Gast, V., L. Bierkandt & C. Rzymiski (2015). 'Creating and retrieving tense and aspect annotations with GraphAnno, a lightweight tool for multi-level annotation'. In Bunt, H. (ed.): *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Annotation*, 23-28. Tilburg: Tilburg Center for Cognition and Communication.
- ▶ Gast, V., L. Bierkandt & C. Rzymiski (2015). 'Annotating modals with GraphAnno, a configurable lightweight tool for multi-level annotation'. In M. Nissim & P. Pietrandrea (eds.): *Proceedings of the Workshop on Models for Modality Annotation*, 19-28. Stroudsburg, PA : Association for Computational Linguistics (ACL).