Corpus annotation

Session 11: 10.01.2024

Fafa Same

Course evaluation

- https://uzk-evaluation.uni-koeln.de/evasys/online.php?pswd=YFAXE

Corpus annotation

What is annotation?

"The process of adding [...] interpretive, linguistic information to an electronic corpus of spoken and/or written language data" (Leech 1997)

Why do we annotate?

- To retrieve and extract information
- To record a linguistic phenomenon explicitly
- To quantitatively study the data
- To compare data or datasets
- To build computational models

Possible research questions

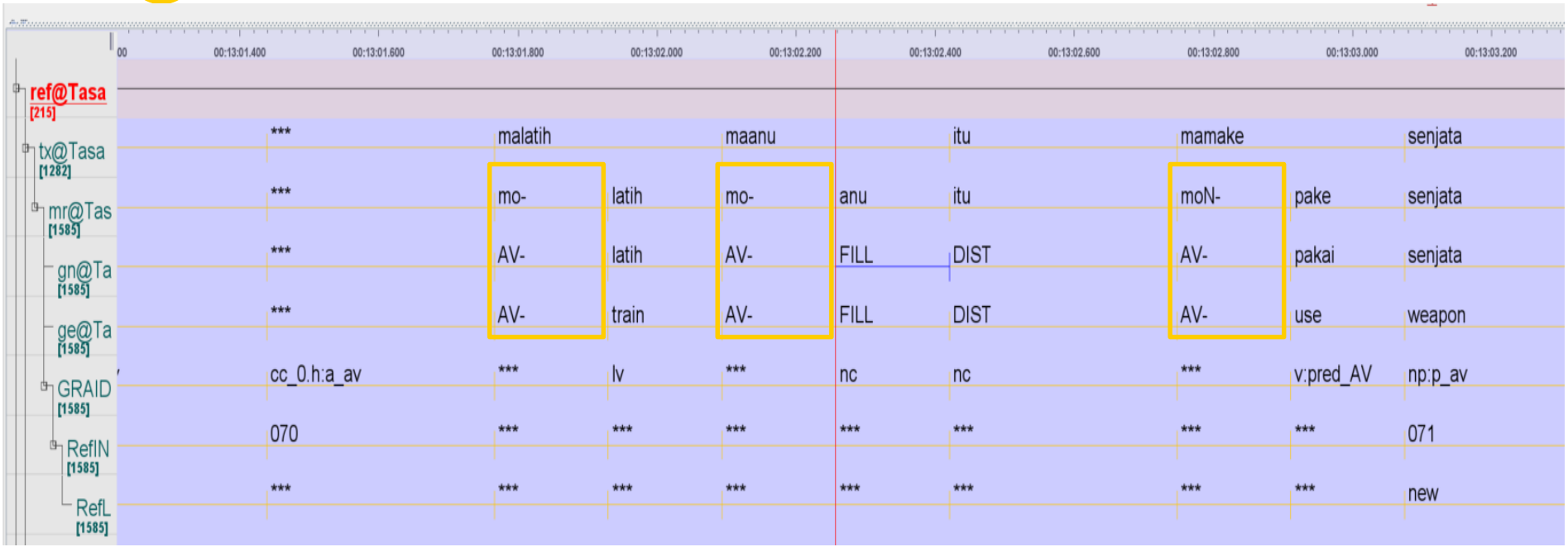
- How frequent are [+human] expressions in different syntactic functions?
- How are new referents realized in comparison to given referents?

How are corpora annotated?

- Manual
- Automatic
- Hybrid (a combination of manual and automatic)
- Crowdsourced

What do we annotate?

Morphology



Corpus of Totoli

Morphology

- Interlinear glossing
  - E.g. Leipzig Glossing Rules (LGR)
- Lemmatization (root or lemma)
  - *woman*_WOMAN | *women*_WOMAN | *woman's*_WOMAN | *women's*_WOMAN
- Tokenization
  - Multiword tokens: *in spite of*
  - Multitoken words: *can't*

Parts of speech tagging

- Specifying the word class membership of word forms
- Motivation: surface word forms often have more than one part of speech (e.g., like)
  → searching for word forms can potentially return *false positives.*

a. we_PPSS always_RB **like_VB** to_TO keep_VB the_AT ball_NN

b. that_CS I_PPSS feel_VB **like_CS** a_AT fool_NN Barth & Schnell 2022:116

- A defined and confined inventory of **tags** that is called a ***tagset***
  - *Brown tagset (87 tags): http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM#bc6*
  - Stuttgart-Tübingen tagset (STTS)

Syntax

- Annotating constituent structure or dependencies
- Treebanks → corpora containing constituency/dependency relations

- Famous examples:
  - Penn treebank
  - Prague Dependency Treebank

Constituency parsing

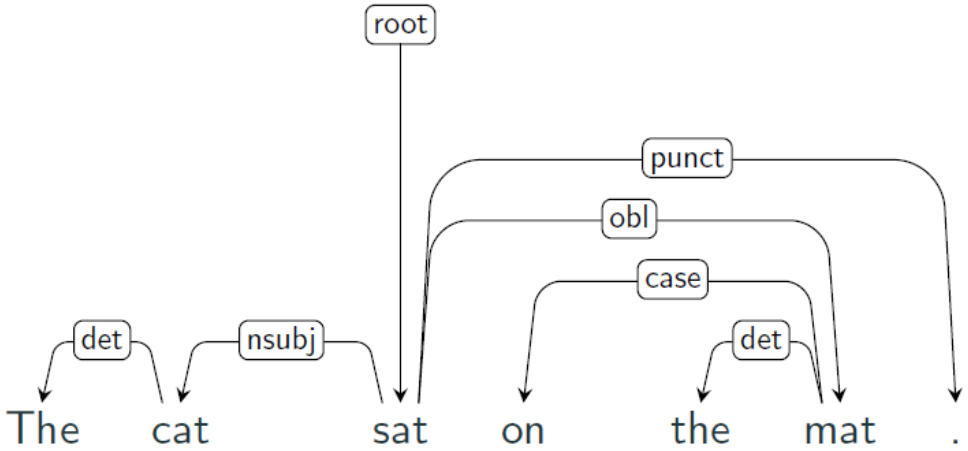
- **Hierarchical** → dividing sentences into their phrasal constituents
- **Bracketing**: a bracket notation (alternative to tree structure) is used to render the underlying hierarchical phrase structure
- Can be enriched with:
  - **functional** annotation → predicate-argument structure
  - POS tagging

```
(TOP (S (NP-SBJ (NNP Mr.)
                (NNP Vinken))
        (VP (VBZ is)
            (NP-PRD (NP (NN chairman))
                    (PP (IN of)
                        (NP (NP (NNP Elsevier)
                                (NNP N.V.))
                            (, ,)
                            (NP (DT the)
                                (NNP Dutch)
                                (VBG publishing)
                                (NN group))))))
        (. .)))
```

OntoNotes corpus (Weischedel et al., 2013)-wsj0001

Dependency parsing

- The sentence syntax is expressed in terms of dependencies between **words** → no use of phrasal constituents.
- Dependencies: directed, typed edges between words in a graph



Semantics

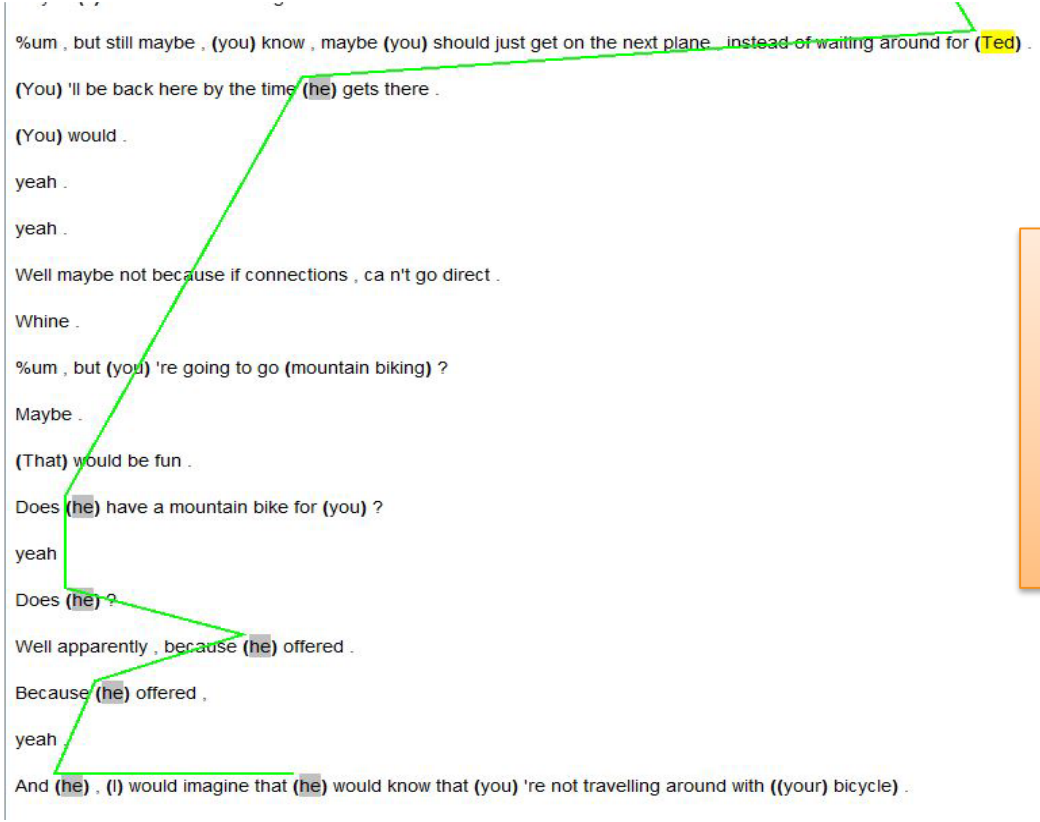
- Word sense disambiguation (e.g., different senses of the word RUN)
- Semantic role labeling (e.g., agent, patient, experiences)
- Named entity recognition (person, organization, time, etc)

```
0   Taiwan
       coref: IDENT          7      0-0     Taiwan
       name:  GPE                   0-0     Taiwan
1   has
2   improved
       sense: improve-v.1
       prop:  improve.01
        v           * -> 2:0,   improved
        ARG0        * -> 0:1,   Taiwan
        ARG1        * -> 3:2,   its standing with the U.S.
        ARGM-MNR    * -> 8:1,   by *PRO*-2 initialing a bila
                                introducing legislation 0 *P
                                showings of their films *T*-
3   its
       coref: IDENT          7      3-3     its
4   standing
       sense: standing-n.1
5   with
6   the
       coref: IDENT          12     6-7     the U.S.
7   U.S.
       name:  GPE                   7-7     U.S.
```

https://verbs.colorado.edu/html_groupings/improve-v.html

Discourse and reference

- *Text* or *discourse: multiple* utterances interrelated with each other in a *coherent way.*
- *Two major types of annotation:*
  - The **coherence** relationships between utterances/propositions (e.g., elaboration, explanation)
  - **Co-reference relations** between the mentions of the same entities in a discourse
    - *Anaphora resolution: which referent*
    - *Referential choice: what is the form of the referring expression*



A text snippet from a telephone conversation in OntoNotes with a referential chain

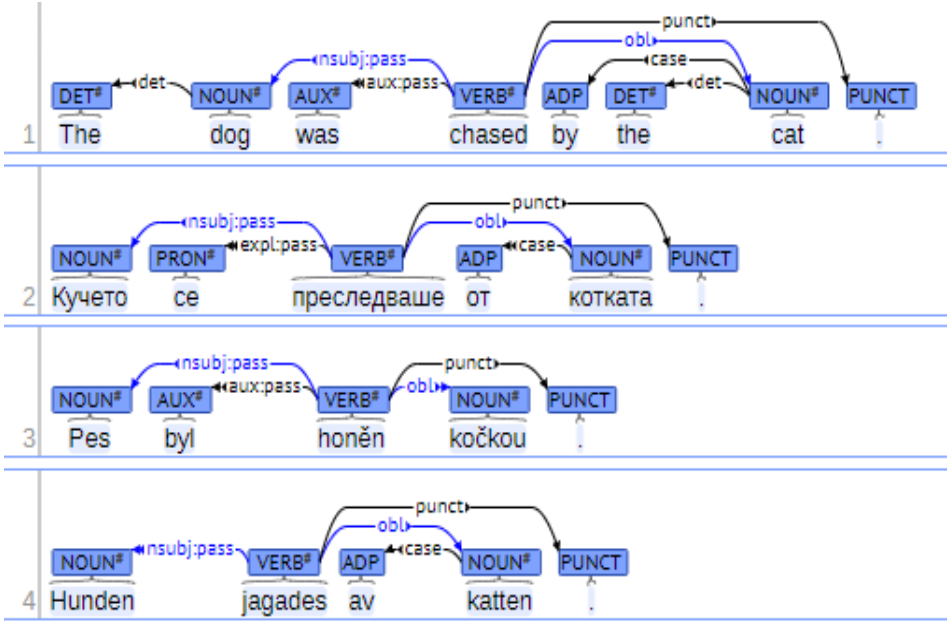
Other categories

- Pragmatics
- Phonetics and prosody
- Extra-linguistic phenomena, e.g., gaze or gesture

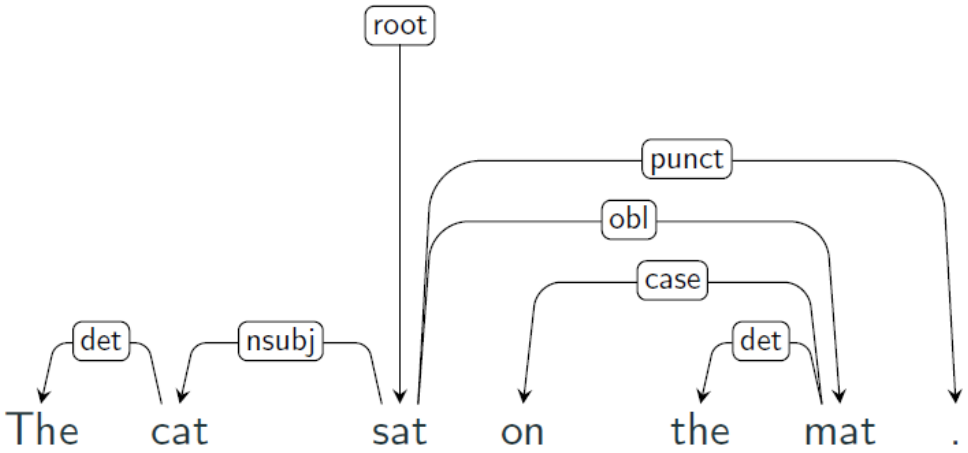
Cross-lingual annotation framework

Universal Dependencies (UD)

- Cross-linguistically consistent treebank annotation
- Universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages.
- Goal: facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective



Dependency parsing



Universal Dependencies (UD)

- Closed set of annotation labels.
- Problems:
  - Heavily based on English/major European languages.
  - Cross-linguistic application.

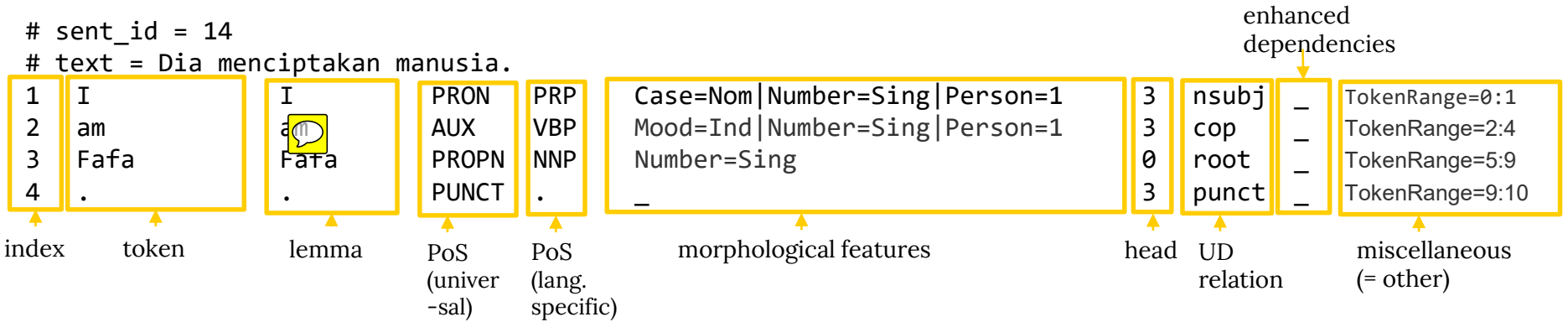
E.g. Universal Dependency Relations

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

What is CoNLL-U format?

- A standard format for text annotation in Universal Dependency
- Widely recognized in the NLP community
- The format uses a tab-separated, column-based structure for each token in a sentence.
- Each token is annotated for different info (10 columns) including ID, form, lemma, Universal POS, language-specific POS, morphological features, dependency relation
- There is also sentence level annotation (marked with #)

Universal Dependencies (UD)



```
# generator = UDPipe 2, https://lindat.mff.cuni.cz/services/udpipe
# udpipe_model = english-ewt-ud-2.12-230717
# udpipe_model_licence = CC BY-NC-SA
# newdoc
# newpar
# sent_id = 1
# text = I am Fafa.
1	I	I	PRON	PRP	Case=Nom|Number=Sing|Person=1|PronType=Prs	3	nsubj	_	TokenRange=0:1
2	am	be	AUX	VBP	Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin	3	cop	_	TokenRange=2:4
3	Fafa	Fafa	PROPN	NNP	Number=Sing	0	root	_	SpaceAfter=No|TokenRange=5:9
4	.	.	PUNCT	.	_	3	punct	_	SpaceAfter=No|TokenRange=9:10
```

Automatic annotation

What can be annotated automatically? (1)

Tokenization: Segmenting text into tokens.

Part-of-Speech (POS) Tagging: Assigning parts of speech to each token.

Lemmatization: Reducing words to their base form.

Named Entity Recognition (NER): Classifying named entities into categories (persons, organizations, etc.).

Morphological Analysis: Analyzing word structures and components

Dependency and Constituency Parsing: Analyzing sentence structure.

Chunking (Shallow Parsing): Breaking up a sentence into syntactically correlated parts of words, like NPs, VPs, etc.

What can be annotated automatically? (2)

Coreference Resolution: Identifying when words refer to the same entity.

Semantic/thematic Role Labeling: Identifying the basic who-did-what-to-whom structure of a sentence.

Sentiment Analysis: Determining the writer's attitude, e.g., positive, negative, or neutral.

Discourse Analysis: Analyzing text structure above the sentence level.

Word Sense Disambiguation: Determining which sense of a word is used in a sentence.

Phonetic and Prosodic Annotation: Analyzing the sound aspects of speech.

Gesture Analysis: Examining the role and meaning of physical gestures in communication.

Automatic annotation solutions

Web Servies

- Weblicht (link) segmentation, tokenization, lemmatization, POS tagging, parsing, etc
- BAS Web service (link) phonetic analysis
- UDPipe (link): dependency parsing

Different Libraries/models

- R: tm (text mining), quanteda, tidytext, spacyr, tokenizers, udpipe
- Python: spaCy, NLTK, StanfordNLP, AllenNLP, CoreNLP
- Huggingface & OpenAI: Whisper, Pyannote, XLNet, BERT, RoBERTa, etc.

Basic NLP pipeline with spaCy


```
# Sample text
text = "Today, we talk about automatic annotation tools."


doc = nlp(text)

data = []

# Process the text
for token in doc:
    data.append({
        "Text": token.text,
        "Lemma": token.lemma_,
        "POS": token.pos_,
        "Tag": token.tag_,
        "Dep": token.dep_,
        "Stop": token.is_stop
    })


df = pd.DataFrame(data)
```

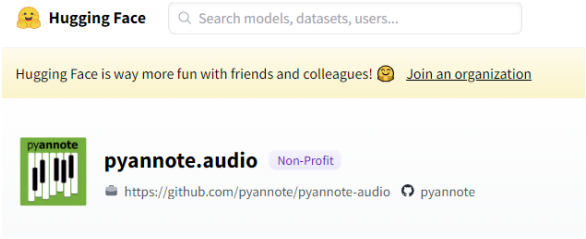
```
[38] print(df)
```

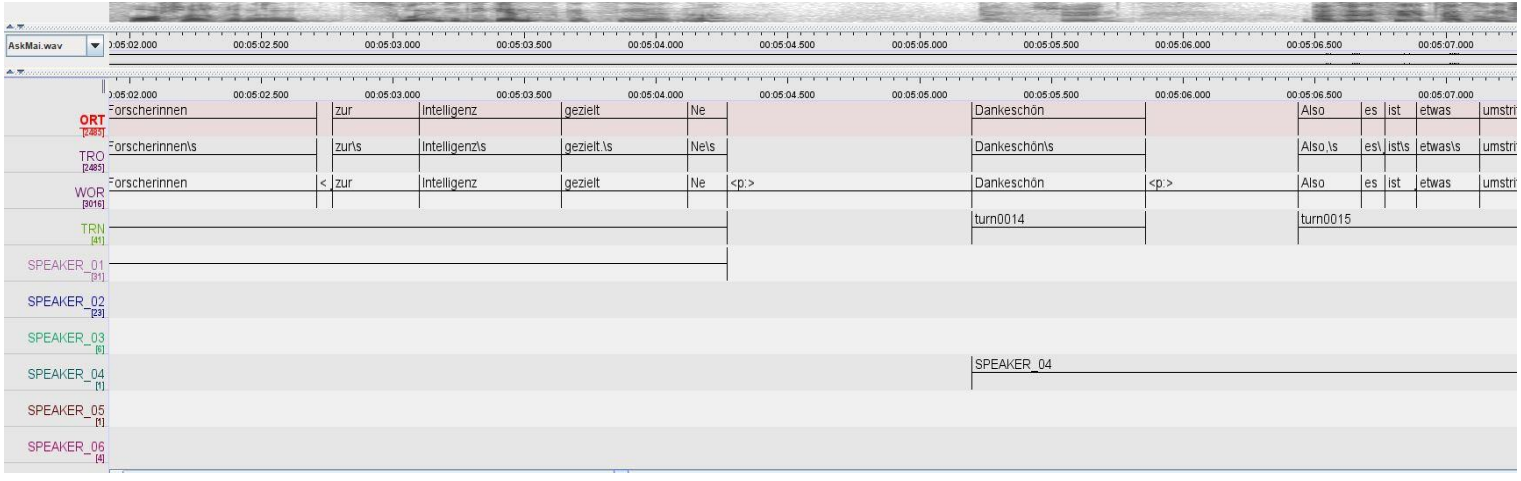
	Text	Lemma	POS	Tag	Dep	Stop
0	Today	today	NOUN	NN	npadvmod	False
1	,	,	PUNCT	,	punct	False
2	we	we	PRON	PRP	nsubj	True
3	talk	talk	VERB	VBP	ROOT	False
4	about	about	ADP	IN	prep	True
5	automatic	automatic	ADJ	JJ	amod	False
6	annotation	annotation	NOUN	NN	compound	False
7	tools	tool	NOUN	NNS	pobj	False
8	.	.	PUNCT	.	punct	False

Automatic Transcription of German Conversations

- Automatic Speech Recognition (ASR) [BAS Web Service, Fraunhofer]

- Diarization: Pyannote [Python script]





CoreNLP for Mention (RE) and chain detection

```
Sentence 0: Jim : Have you heard the news about Maria Williams ?
Sentence 1: Mary : Who ?
Sentence 2: Jim : The genius in charge of ABC company .
Sentence 3: Mary : Right !
Sentence 4: the famous Williams .
Sentence 5: What about her ?
Sentence 6: Jim : She decided to quit ABC and already announced Nina Becker as her successor .
Sentence 7: Mary : You kidding me !
Sentence 8: Nina is not fit for this job .
Sentence 9: Jim : I know .
Sentence 10: I 'm afraid she 's going to ruin all Maria 's legacy .


Chain 0:
        Mention 0: Referring Expression='Jim',  sentence 0, tokens 0-0
        Mention 1: Referring Expression='you',  sentence 0, tokens 3-3
        Mention 2: Referring Expression='Jim',  sentence 2, tokens 0-0
        Mention 3: Referring Expression='Jim',  sentence 6, tokens 0-0
        Mention 4: Referring Expression='You',  sentence 7, tokens 2-2
        Mention 5: Referring Expression='Jim',  sentence 9, tokens 0-0

Chain 1:
        Mention 0: Referring Expression='Nina Becker',  sentence 6, tokens 10-11
        Mention 1: Referring Expression='Nina',  sentence 8, tokens 0-0

Chain 2:
        Mention 0: Referring Expression='me',  sentence 7, tokens 4-4
        Mention 1: Referring Expression='I',  sentence 9, tokens 2-2
        Mention 2: Referring Expression='I',  sentence 10, tokens 0-0

Chain 3:
        Mention 0: Referring Expression='Maria Williams',  sentence 0, tokens 8-9
        Mention 1: Referring Expression='the famous Williams',  sentence 4, tokens 0-2
        Mention 2: Referring Expression='her',  sentence 5, tokens 2-2
```

OpenAI Whisper + GPT API integration

- Whisper

Transcript = 'Hier ist ein Beweis dafür, dass es auch gute deutsche Serien gibt. Da hätten wir zum einen Legal Affairs, das ist einfach wie das deutsche Suits, mega geil. ,

Translate = 'Here's a proof that there are also good German series. There we have Legal Affairs, which is just like the German Suits. ,

- Further GPT integration

Word - **Translation** - **Transcription**:

Hier - here - hɪr
ist - is - ɪst
ein - a - aɪn
Beweis - proof - bəvaɪs
dafür - for that - dafʏːɐ̯
dass - that - das
es - it - ɛs
auch - also - aʊx
gute - good - guːtə
deutsche - German - dɔʏtʃə
Serien - series - zeˈʁiːən
gibt - there are - gɪpt