Coconstructions in Spoken Data: UD Annotation Guidelines and First Results

With a short introduction to WG1's work on spoken data annotation

Ludovica Pannitto Sylvain Kahane Kaja Dobrovoljc Zor
Elena Battaglia Bruno Guillaume Caterina Mauri Eleonora Zucchini

University of Bologna

Université Paris Nanterre, Modyco

University of Ljubljana / Jozef Stefan Institute

Université de Lorraine, CNRS, Inria, LORIA

Masaryk University, Brno

UniDive WG1 T1.5 – Bucharest, 2026

Outline

1 Spoken Data and Task 1.5

2 Coconstructions in Spoken Data

3 Treebank Organization

4 Conclusion

What is a spoken treebank?

- All languages are spoken, over 40% of them have no written form at all, yet speech is under-researched as a mode of human communication
- A spoken corpus is a collection of **transcribed speech recordings**
  - It can represent different types of communicative event (monologic narration; interview; free conversation; podcast; news broadcast...)
  - The recording may be available, and may be aligned with the transcription
  - The transcriptions may include layers of annotation (PoS; lemmatization; dependency relations...)
  - Usually a considerable amount of metadata is part of the data
- A spoken treebank can be considered as an additional layer of annotation added on top of a spoken corpus
- Syntax can act as an entry point to much more information (semantic, pragmatic, interactional...), making the treebank a hub for further linguistic analysis

Peculiarities of spoken language

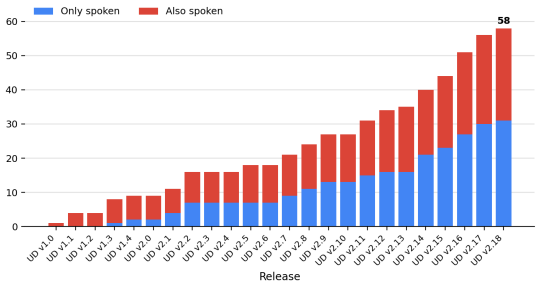
- Unlike written language, transcriptions are not the native data format but rather the outcome of a series of **explicit descriptive choices**
- Not only monological, conversations may be multi-party (overlaps; co-constructions)
- No agreement about what a sentence is in spoken language



- Speech contains additional phenomena in comparison to written language, which can be considered part of the linguistic material:
  - disfluencies, hesitations, false starts, self-repairs
  - interruptions
  - backchannels (*mhm*, *yeah*, *right* . . .)
- Many formal annotation attempts, but no shared conventions

Spoken data in Universal Dependencies

- Spoken data are no longer marginal
- **58 treebanks** in UD v2.18 contain spoken data
  - 31 contain spoken data only, for a total of over **93,000 sentences** and **870,000 tokens**
- They cover **49 languages** from **31 language families**
- For 17 languages, the spoken treebank is the **only UD treebank** available



UniDive WG1 task on spoken data annotation

- **Objectives**
  - Identify and discuss issues related to spoken data annotation
  - Propose harmonized guidelines for the annotation of speech-specific phenomena
  - Promote awareness of spoken data in other WG1 activities and tasks

- **Two main outcomes**
  1. A survey of how spoken data is currently handled in (S)UD
     - already presented at GM3, Budapest 2025
     - main finding: many speech-specific phenomena, few shared practices
  2. A proposal for common guidelines
     -  a first proposal, openly available and still growing

Towards shared guidelines for spoken data in UD

- Regular task meetings comparing existing practices and discussing solutions, phenomenon by phenomenon

- Recommendations at several levels, among them:
  - **Unit segmentation:** what counts as a sentence in speech?
  - **Speech-specific metadata**: common inventory and naming conventions
  - **Speech-specific phenomena**: self-repairs, discourse markers, coconstructions ...

→ **Today: Coconstructions as an illustrative use case**: from discussion to guidelines to cross-lingually harmonized data

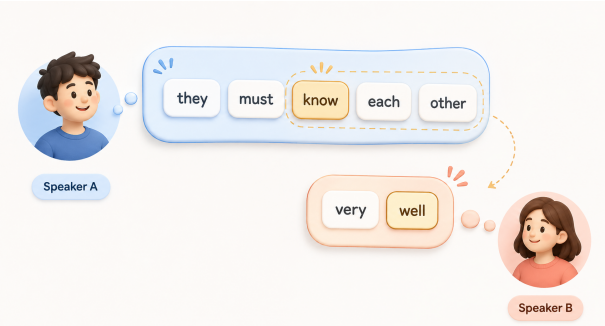


Figure: Image produced by ChatGPT, example from Ono & Thompson (1996)

How many **sentences**?

- two turns
- two (or maybe more) prosodic units
- one rectional unit

Two units of analysis for spoken UD treebanks

Speaker-dependent maximal units

Loosely related to the notion of *turn*; anchor segmentation to each speaker's production

Structurally-autonomous maximal units (*rectional units*)

Based on the notion of **government** (Tesnière 1959; Blanche-Benveniste 1987; Kahane & Pietrandrea 2012): defined by words imposing constraints on other words (category, morphosyntax, linear position)

A single rectional unit may be **jointly realized** across turns, by one or more speakers → **coconstruction**

Coconstructions: definition

Coconstruction

A syntactic relation (any dependency type) that holds between utterances produced by (i) **different speakers**, or (ii) the **same speaker** after an interruption by another participant

- Speakers may jointly build a single syntactic structure, in a coordinated or overlapping fashion (Lerner 1991; Ono & Thompson 1996; Sacks 1992)
- Grounded in the notion of *unité rectionnelle* (Benzitoun et al., 2010; Kahane & Pietrandrea, 2012)
- Three major types: **coconstruction proper**, **wh-question** answers, **backchannelling**

Type 1: Coconstruction proper

Multiple contributions target the **same dependency structure**: completing, duplicating, or extending it.

A. Completion of a projected dependency

English (Ono & Thompson, 1996):
L: *his position is pretty uh* A: ***stable***

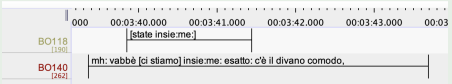
B. Extension: a new dependency is added

English (Ono & Thompson, 1996):
M: *they must know each other* H: ***very well***

Type 1 (cont.): parallel and subsequent realization

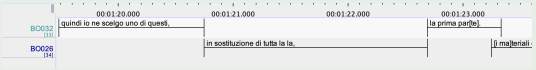
Parallel realization

Italian (KIParla, BOD2018) — both speakers fill the same slot near-simultaneously:
BO140: *mh: vabbè [ci stiamo] insie:me:* 'well we stay together there' BO118: ***[state insie:me:]*** 'stay together'



Subsequent realization / repair

Italian (KIParla, BOA1003):
BO032: *quindi io ne scelgo uno di questi, . . . la prima par[te].* 'so I choose one of these . . . the first part' BO026: ***in sostituzione di tutta la la, . . . [i ma]teriali*** 'in place of all of the . . . materials' (`deprel = repair`)



Type 2: Wh-question answers

- The answer to a wh-question is analysed as **syntactically dependent** on the question → part of the same rectional unit
- Already proposed in the Rhapsodie guidelines

Example (Italian, KIParla)

TOR001: ***dove*** *vai ad arrampicare?* 'where do you climb?'
TOI007: ***al Bside*** *vicino alla colletta* 'at the Bside, close to the belay'

Not a coconstruction: polar questions

Answers to yes/no questions do **not** fill a pre-existing syntactic slot ⇒ not treated as coconstructions

Type 3: Backchannelling

- Short (verbal/paraverbal) productions by one participant while another holds the floor: *mhm*, *yeah*, *right* ...
- Function: signal attention, alignment, understanding — not new syntactic material
- Criteria to qualify as a backchannel:
  1. addressed to the content of the other speaker's contribution
  2. **not** required/expected by the preceding turn (unlike wh-answers)
  3. does not require a reaction from the main speaker
- Encoded via `MISC` without altering the dependency graph — makes this interactional layer visible without disrupting existing annotation

Proposed guidelines: MISC features

Two new `MISC` features, fully backward-compatible (core dependencies, POS, tokenization untouched):

`Coconstruct`

Backward pointer ⟨`deprel`⟩`::`⟨`sent_id`⟩`::`⟨`tok_id`⟩ to the eliciting/projecting token

`Backchannel`

Backward pointer ⟨`sent_id`⟩`::`⟨`tok_id`⟩ (relation is always `discourse:backchannel`)

Assigned to the token that comes **second in time** (the responding token), pointing back to the earlier anchor.

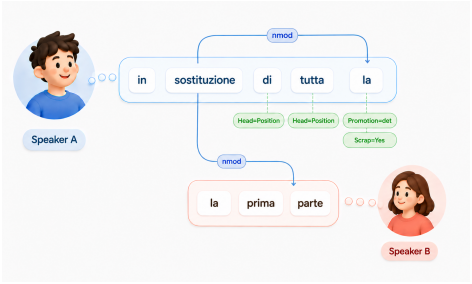
First configuration for `Coconstruct`

Completion of an open dependency slot

- Speaker 1 projects a dependent that is never realized
- Speaker 2's token realizes it
- Governor = the projecting token (standard UD head rules)

Handling unfinished structures

- Problem: Speaker 1 leaves a dependency **unrealized** (e.g. a determiner with no following noun) → UD needs a well-formed tree
- `Scrap=Yes` – marks an element/subtree whose syntactic projection remains unfinished
- `Promotion=`⟨`deprel`⟩ – records the relation the element would bear to its governor if realized
- `Head=Word/Position` – resolves ambiguity when a dependent of the promoted word could attach to the promoted word itself or to its missing head



Second configuration for `Coconstruct`

Stacking on an already realized slot

- Speaker 1's structure is already complete
- Speaker 2 adds an **alternative** in the same slot
- Reformulation (`conj:reform`), including answers to wh-questions

New relation for reformulation: `conj:reform`

- Standard `conj`: conjuncts have **different referents**
- `conj:reform`: conjuncts are **different denotations of the same referent**
- In spoken data it covers: confirmations, wh-question answers (alternative case), corrections/reformulations across or within turns

From speaker-based to dependency-based views

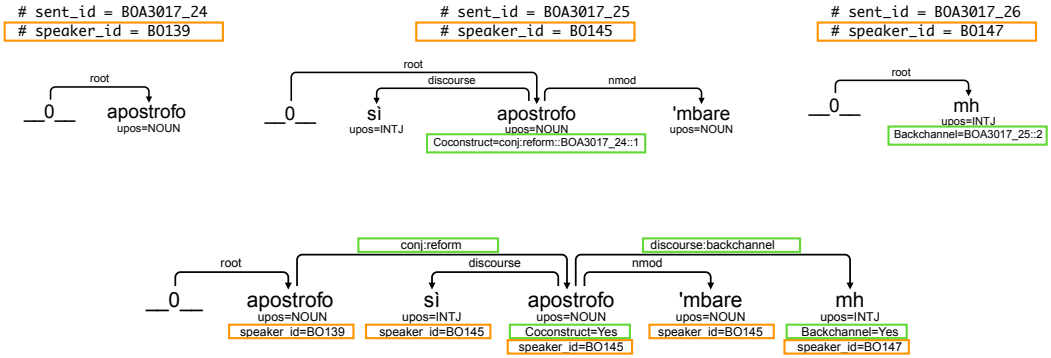
- We propose two UD representations:

  Speaker-based view maximal units are built according to "if you can link, link", but cannot extend across a change of speaker; cross-speaker dependencies are recorded through `Coconstruct`.

  Dependency-based view segmentation follows **rectional units**; dependencies across speakers are represented as regular dependencies within the same structure.

- The two views are connected through an **automatic conversion**.
- Speaker metadata moves from sentence level to **token level**, since one dependency-based sentence can span several speakers
- Units identified in the source corpus—e.g. illocutionary or intonation units—belong to an orthogonal layer and may be preserved as metadata when available.

Speaker-based to dependency-based example



Coconstructions annotated in spoken UD treebanks (2.18)

Treebank	Language	Coconstruction annotation
Rhapsodie	French	3,209 sentences; 315 backchannels; 81 coconstructions
KIParla Forest	Italian	2,221 sentences (18,680 tokens); 309 backchannels; 123 coconstructions
SST	Slovenian	6,121 utterances; 386 semi-automatically identified backchannels; 17 manually confirmed coconstructions

Beyond spoken interaction

- The framework generalizes to **computer-mediated communication** (CMC): instant messaging, Slack/Discord, where users fragment one syntactic unit across several messages ("split utterances")
  - `Coconstruct`, `Scrap`, `Promotion` apply directly
  - Pointer-based design is robust to non-adjacent replies (message *lag*)
- Also useful for **same-speaker** continuation across a long backchannel sequence, without collapsing turn segmentation

Two organizational problems

Loss of original document structure

Standard CoNLL-U train/dev/test splits do not preserve the **original document divisions** of the source corpus (cf. UD issue #1146 and #1282)

Metadata redundancy

Document- and speaker-level information is currently **repeated across every sentence** → maintenance burden, reduced readability

Flexible File Organization

In order to keep original document structure:

- Documents can optionally live in a `not-to-release/original_split` subfolder
- A `merge.json` file describes how to **reconstruct** the final UD files from the original documents/sentences
- Standard `dev`/`test`/`train` splits are still produced at the repository root
- A `document_id` metadata added with the original filename.

```
{ "fr_rhapsodie-ud-test":
  [
    { "sent_ids": ["Rhap_M0001-9", "Rhap_M0001-10", ...] },
    { "document_ids": ["Rhap_D2011", "Rhap_D2010", ...] }
  ]
}
```

Metadata Sharing

- A `metadata.json` file stores metadata **once**, when one value deterministically implies another
- e.g. "for each `speaker_id`, there is exactly one `speaker_age`"

```
{ "speaker_id": {
    "§LF4": { "speaker_age": "20" }
  }
}
```

Common dependencies in spoken corpora:

- `speaker_id` → `speaker_age`, `speaker_sex`
- `document_id` → `sound_url`

Adoption and tooling

Treebanks implementing the solutions in 2.18

- `metadata.json`: 3 treebanks: UD_French-ParisStories, UD_French-Rhapsodies and UD_Italian-KIParlaForest
- `merge.json`: 12 treebanks (converted from SUD): Beja, French, Haitian Creole, Hausa, Nenets, Zaar.

Supporting scripts (`UniDive/SpLAn-UD` github repository):

- `merge_and_unshare.py` – rebuilds final UD files, expanding shared metadata
- `metadata_share.py` – extracts shareable metadata to reduce repetition
- `metadata_detect_shareable.py` – discovers dependencies automatically

Summary

- Spoken language challenges UD's implicit assumption that syntax operates within sentence boundaries
- **Coconstructions** let us represent syntactic dependencies that cross speaker turns, via lightweight, backward-compatible `MISC` features (`Coconstruct`, `Backchannel`, `Scrap`, `Promotion`) and a new `conj:reform` relation
- Speaker-based and dependency-based views can be kept separate and **automatically converted**
- Source-defined units remain an **optional and orthogonal layer**: existing information can be preserved where available
- Complementary guidelines on **metadata** and **treebank organization** improve consistency, preserve source-corpus structure, and reduce redundancy
- Framework extends naturally to **CMC data** and other interactional settings

Future work

- Keep the spoken treebanks group active: plan regular meetings approx. 1 month before official UD releases to harmonize resources
- Creation of a dedicated mailing list for T1.5 and **whoever wants to join** to work on spoken treebanks
- Work on harmonization of metadata – see the table on guidelines proposal
- Plan new issues to discuss (e.g., how to use punctuation, better encoding of non verbal phenomena etc.)

Outlook

- Work of **UniDive WG1 Task 1.5**, supported by COST Action CA21167 and ARIS programme P6-0411
- Goal: submit this proposal to the UD community as a **standard extension** for spoken and interactive treebanks
- Toward a more linguistically adequate treatment of **multi-party and mediated** language data

Thank you!