Journal of Open Humanities Data

Reuse by Design: A Pivot-Based Architecture for the KIParla Corpus of Spoken Italian

DISCUSSION PAPER

LUDOVICA PANNITTO
CATERINA MAURI
*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

KIParla is a large, modular corpus of spontaneous spoken Italian originally transcribed in ELAN using Jefferson-style conventions. While this representation preserves fine-grained interactional detail and time alignment, it limits interoperability, large-scale querying, and computational reuse. This paper presents the design and implementation of a pseudo-tokenized, verticalized pivot format developed to support validation, maintenance, and infrastructural integration without sacrificing descriptive richness. The proposed format makes explicit the analytical units implicit in Jefferson transcription—transcription units, spans, and tokens—and enforces well-formedness constraints at character, span, and unit levels. Overlap, the most complex relational phenomenon, is resolved through a graph-based algorithm that derives temporal overlap events from alignment data and deterministically matches them to textual spans. Each token is represented as a structured record enriched with lexical, prosodic, interactional, and alignment features, anchored through explicit character offsets. The vertical format functions as a maintained pivot representation from which alternative formats, including ELAN files and UD-compatible treebank representations, can be reproducibly derived. This architecture enables large-scale lemmatization, part-of-speech tagging, syntactic annotation, and cross-layer querying, while supporting version-controlled, DevOps-inspired workflows for sustainable corpus growth. The KIParla pivot format thus reconciles interaction-oriented transcription practices with computational standards and provides a model for reuse-oriented spoken-language data engineering.

CORRESPONDING AUTHOR:
Caterina Mauri
Department of Modern languages, literatures and cultures, University of Bologna, Bologna, Italy
caterina.mauri@unibo.it

KEYWORDS:
spoken language corpora; interactional annotation; data interoperability; version-controlled workflows; universal dependencies

TO CITE THIS ARTICLE:
Pannitto, L., & Mauri, C. (2026). Reuse by Design: A Pivot-Based Architecture for the KIParla Corpus of Spoken Italian. *Journal of Open Humanities Data*, 12: 81. DOI: https://doi.org/10.5334/johd.527

1 CONTEXT AND MOTIVATION

Human language is first and foremost spoken, yet spoken interaction remains structurally underrepresented in language resources across the humanities and computational linguistics (Chrupała, 2023; Dobrovoljc, 2022; Linell, 2019). While written corpora benefit from relatively stable formats, mature annotation standards, and established querying practices, spoken data pose persistent challenges of collection, annotation, representation, and reuse—especially when interactional phenomena (overlaps, repairs, co-construction, disfluencies, backchannels, discourse particles) are treated as marginal or excluded from downstream formats.

This paper investigates how the KIParla corpus (Mauri et al., 2019) can be reused, reformatted, and extended to improve interoperability without sacrificing the richness of interactional transcription and annotation. We address not only technical aspects—releasing KIParla in a new, lightweight, highly interoperable format—but also conceptual issues of text segmentation, unit identification, annotation stability, versioning, and relations between representational layers.

By situating KIParla within this broader methodological and infrastructural landscape, the paper contributes to debates on best practices for developing and reusing spoken language datasets in the humanities. We treat reuse not as a simple technical task but as a process that makes explicit the assumptions, trade-offs, and epistemic commitments embedded in data formats and standards, arguing that spoken corpora such as KIParla serve both as empirical resources and as testbeds for rethinking interoperability, annotation, and reuse in language-focused research.

1.1 THE KIParla CORPUS

The KIParla corpus was conceived to provide a large-scale, carefully constructed collection of spontaneous spoken Italian. Initiated in 2016 and developed through the years as a collaboration between the Universities of Bologna and Turin, KIParla was designed from the outset as a modular and extensible corpus, with new modules added over time following shared transcription conventions and compatible metadata structures. The corpus currently comprises six released modules - KIP (Goria & Mauri, 2018), KIPasti (Mauri et al., 2024a), ParlaBO (Mauri et al., 2024b), ParlaTO (Cerruti & Ballarè, 2021), StraParlaBO and StraParlaTO (Mauri et al., 2025) - and several under development in collaboration with other Universities (University of Bozen, Naples, Munich), covering a wide range of interactional settings such as free conversations, semi-structured interviews, lectures, oral exams, and office hours, recorded in different Italian regions and social contexts. Rich speaker and conversation metadata support sociolinguistic and interactional analyses across age, occupation, educational background, and communicative situation. As of today, the resource counts ca. 350 hours of recorded and transcribed conversation for a total of approximately $3,200,000$ tokens.

The architecture of KIParla is grounded in a principled combination of modularity and incrementality. Each module is designed as internally balanced, constructed according to explicit sampling criteria and sociolinguistic parameters, and can be queried independently as a self-contained dataset. The modules are however structurally interoperable, sharing annotation standards, metadata schemes, and design principles that allow them to be searched jointly. When considered separately, each module supports fine-grained analyses within a controlled domain; when interrogated together, the corpus provides a broader empirical space in which patterns can be compared across contexts, speaker profiles, and interactional settings.

This dual architecture is what makes KIParla capable of sustained growth without sacrificing methodological rigor. Crucially, expansion does not dilute balance or comparability, because every new module enters the system as a structured and calibrated unit.

1.2 THE INPUT DATA: LIMITATIONS AND POTENTIALITIES

The KIParla pipeline (Figure 1) has been followed for each published module. It starts with data collection: a shared process brought about by a community of workers, including the PI responsible for the module, various researchers up to students and interns who actively take part in the fieldwork. Such a collective effort continues in the upcoming phases: namely, archiving and transcribing. The corpus is orthographically transcribed and enriched with a set of transcription conventions inspired by Jefferson (2004), a widely adopted system in Conversation Analysis due to its expressive power and ability to capture fine-grained interactional detail.

Through ELAN (Max Planck Institute For Psycholinguistics, The Language Archive, 2025; Wittenburg et al., 2006), the audio recordings are transcribed into speaker-dependent units, aligned with the audio (`.wav`) input.

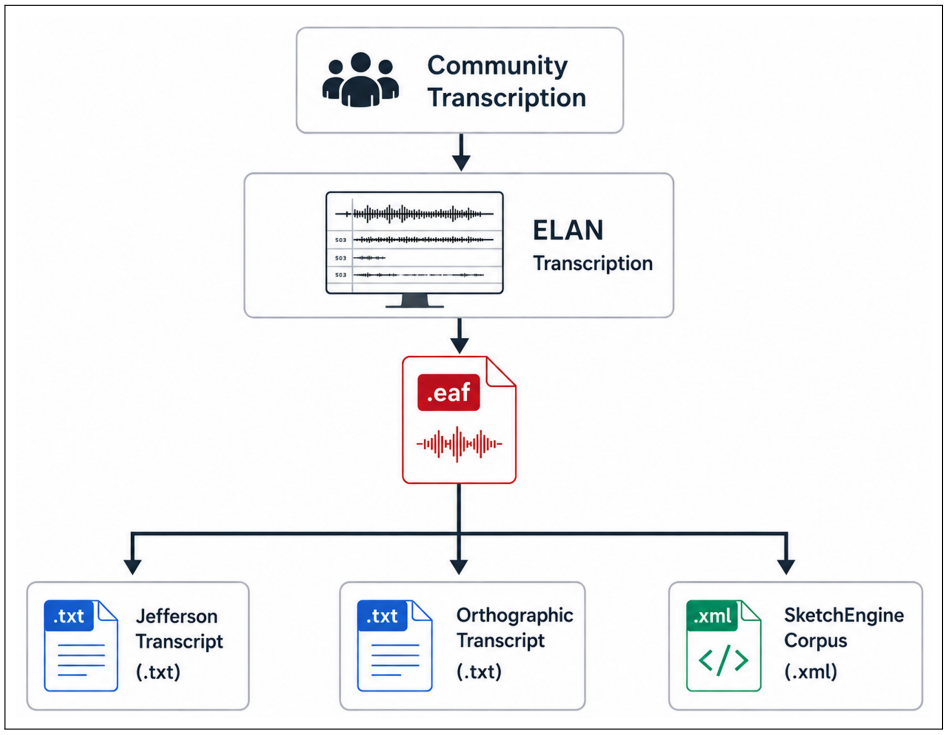

Figure 1 Current KIParla pipeline: data processing phases.

A single expert linguist is then in charge of the overall revision and pseudonymization, to guarantee an acceptable level of internal consistency in the workflow, despite the many different people that intervened during the process, and the difficulties in undertaking extensive and systematic inter-annotator agreement monitoring (see Artstein and Poesio (2008); Gagliardi (2018) for a critical discussion).

The outcome of the data processing phases is an `eaf` file: a textual XML object to be interpreted only by means of the ELAN software (Figure 2). From this, in order to make the data readable and partially queriable through concordancers such as noSketchEngine (Institute of the Czech National Corpus, 2025), a number of linearized representations have to be derived. For instance Figure 3 shows that the orthographic representation derived from the corresponding Jeffersonian one deletes overlaps (e.g., '[no no no]' > 'no no no'), prolongations (e.g., 'venuta::' > 'venuta'), intonation patterns (e.g., 'colloquio,' > 'colloquio').

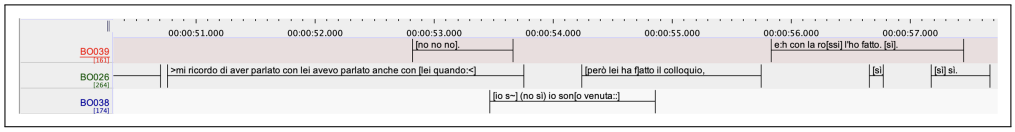

Figure 2 Screenshot of ELAN showing the alignment between the audio (`.wav`) file and the transcription units for file BOA1010 in KIP.

The full `eaf` transcriptions come with some limitations: (i) they are not natively compatible with widely used computational formats and standards, (ii) the format does not come with clear unit descriptions, (iii) the process that brings data to this output is very difficult to scale, and (iv) it is very difficult to revert decisions that have been taken along the way. The lack of a clear multilayer annotation format makes it hard to reuse, automate or investigate aspects independently from one another.

KIParla faces a more general problem in spoken language research: annotation pipelines are often opaque, time-consuming, and difficult to validate, and the resulting data products tend to crystallize ad-hoc analytical choices that are hard to revise, extend, or align with other resources. At the same time, as interests in modeling language in interaction grows both in the humanities and in natural language processing, the need for fully machine readable representations grows as well. Yet, such representations must be able to fully preserve the richness of the original annotation and avoid reducing the data to out of context sentence-level representations.

```
   1  <?xml version='1.0' encoding='UTF-8'?>
   2  <ANNOTATION_DOCUMENT ...>
   6    <TIME_ORDER>
         ...
  13      <TIME_SLOT TIME_SLOT_ID="ts8"  TIME_VALUE="7677" />
  14      <TIME_SLOT TIME_SLOT_ID="ts9"  TIME_VALUE="9586" />
  15      <TIME_SLOT TIME_SLOT_ID="ts10" TIME_VALUE="7861" />   <!-- starts INSIDE ts8..ts9 -->
  16      <TIME_SLOT TIME_SLOT_ID="ts11" TIME_VALUE="8066" />
  17      <TIME_SLOT TIME_SLOT_ID="ts12" TIME_VALUE="8761" />
         ... [1188 more TIME_SLOT entries]
1205    </TIME_ORDER>
1206    <TIER TIER_ID="BO039" LINGUISTIC_TYPE_REF="default-lt">
         ...
1218      <ALIGNABLE_ANNOTATION ANNOTATION_ID="a5"
1219            TIME_SLOT_REF1="ts8" TIME_SLOT_REF2="ts9">
1220        <ANNOTATION_VALUE>ci [so]no altr[e::]</ANNOTATION_VALUE>
1221      </ALIGNABLE_ANNOTATION>
         ...
         ... [~2100 lines: remainder of TIER BO039, then full TIER BO026]
3335    <TIER TIER_ID="BO038" LINGUISTIC_TYPE_REF="default-lt">
3337      <ALIGNABLE_ANNOTATION ANNOTATION_ID="a6"
3338            TIME_SLOT_REF1="ts10" TIME_SLOT_REF2="ts11">   <!-- overlaps a5 -->
3339        <ANNOTATION_VALUE>[no].</ANNOTATION_VALUE>
3340      </ALIGNABLE_ANNOTATION>
3341      <ALIGNABLE_ANNOTATION ANNOTATION_ID="a7"
3342            TIME_SLOT_REF1="ts12" TIME_SLOT_REF2="ts13">   <!-- overlaps a5 -->
3343        <ANNOTATION_VALUE>[e::h credo c]he siano altre quattro...</ANNOTATION_VALUE>
3344      </ALIGNABLE_ANNOTATION>
```

```
1  BO026  >mi ricordo di aver parlato con lei avevo parlato anche con [lei quando:<]
2  BO039  [no no no].
3  BO038  [io s~] [no sì] io son[o venuta::]
4  BO026  [però lei ha flatto il colloquio,
5  BO039  e:h con la ro[ssi] l'ho fatto. [sì].
6  BO026  [sì]
7  BO026  [sì] sì.
```

```
1  BO026  mi ricordo di aver parlato con lei avevo parlato anche con lei quando
2  BO039  no no no
3  BO038  io s~ no sì io sono venuta
4  BO026  però lei ha fatto il colloquio
5  BO039  eh con la rossi l'ho fatto sì
6  BO026  sì
7  BO026  sì sì
```

Figure 3 The same excerpt from conversation BOA1010 in three representations. *Top*: abridged EAF/XML source of BOA1010 (line numbers from the actual file). Time slots ts8–ts12 define intervals that are interleaved in time (ts10 = 7861 ms falls inside ts8–ts9 = 7677–9586 ms), yet the annotations that reference them — annotation a5 in tier BO039 (line 1218) and annotations a6–a7 in tier BO038 (lines 3337–3343) — are separated by over 2100 lines of XML, connected only through opaque time-slot identifiers. *Bottom left*: Jeffersonian linearization. *Bottom right*: orthographic linearization.

1.3 MOTIVATION: A PIVOT FORMAT FOR REUSE-ORIENTED WORKFLOWS

Spoken language research is characterized not by a lack of standards, but by the coexistence of partially incompatible ones, each optimised for specific stages of the data lifecycle. Formats such as `.chat` (MacWhinney, 2019) and `.eaf` (Wittenburg et al., 2006) have proven highly effective for transcription and interactional analysis. However, neither is designed to function as a stable operational format for large-scale linguistic annotation, querying, and infrastructural integration. In both cases, additional annotation layers tend to be external or tool-dependent, limiting extensibility and interoperability across workflows.

The TEI-based ISO standard "Transcription of spoken language" (ISO 24624:2016) (Hedeland & Schmidt, 2022; Werthmann, 2025) addresses structural heterogeneity by providing a shared exchange model grounded in the TEI Guidelines. Its macro-structure supports participants, timelines, and speaker contributions, while allowing theory-dependent variation at the micro-level. ISO 24624 is highly effective as a documentation and conversion standard, facilitating preservation and interoperability across projects. However, it is rarely used as an operational working format for continuous dataset development. In practice, it functions as a target of conversion rather than as the maintained source representation through which annotation, validation, and querying are routinely performed.

Reuse-oriented workflows such as those envisioned for KIParla require a different kind of representation: one that is lightweight, line-oriented, and directly compatible with token-based query systems. In particular, a pivot format should (i) support version control and diff-based inspection, (ii) allow straightforward integration of additional annotation layers, and (iii) serve as a single maintained source from which alternative representations can be reproducibly derived.

A further architectural limitation concerns the hierarchical nature of XML-based formats. XML enforces a strictly nested structure, whereas spoken interaction frequently involves non-hierarchical phenomena such as overlapping speech. While such phenomena can be represented indirectly through references or stand-off annotation, they do not exist as first-class structural objects within the tree model. For reuse scenarios requiring explicit, manipulable representations of relational phenomena, this constraint becomes significant.

The pivot format introduced for KIParla addresses these issues by adopting a vertical, pseudo-tokenized, tab-separated structure in which each row corresponds to a validated transcription-derived token. This format functions as an operational core: it mediates between interaction-oriented transcription practices and computational infrastructures, enabling continuous validation, version-controlled maintenance, and reproducible derivation of alternative representations. Crucially, it has been designed to allow researchers to inspect and understand the data directly, using nothing more than a text editor or spreadsheet software, while remaining fully compatible with automated processing. Basic transformations, validation steps,

and exploratory analyses can be performed using standard command-line tools (e.g., `grep`, `awk`, `sed`) or widely available applications such as spreadsheet editors, without the need for specialized tooling. This emphasis on co-readability reflects a reuse-oriented design philosophy: human readability supports quality control and fast prototyping, while machine readability enables automation, reproducibility, and integration with infrastructure services.

It is important to stress that the pivot format is not intended to replace XML-based standards such as TEI/ISO 24624, which remain essential for archiving, long-term preservation, and cross-project interoperability. Rather, it is conceived as an operational complement suited to the *active development phase* of the corpus, when data are frequently revised, validated, and extended. During this phase, the line-oriented, tab-separated structure offers a decisive practical advantage over XML: whereas XML requires specialized parsers, schema-aware editors, and considerable technical expertise to inspect or modify safely, the pivot representation can be opened, read, queried, and corrected directly in any text editor or spreadsheet application. A researcher can perform quality-control checks or isolate specific annotation layers using a single command-line instruction (e.g., `grep`, `awk`) without relying on XML toolchains. These properties make the pivot format specifically suitable as a maintained working representation during corpus development, while XML-based formats retain their role as targets for archival conversion and interoperability exchange.

2 DATASET DESCRIPTION

The details of the four KIParla modules that have been deposited are provided in Table 1.

Table 1 Metadata and repository information for the deposited KIParla transcript datasets.

KIP transcripts	
Repository location	DOI: https://doi.org/10.60760/unibo/kip Persistent identifier (Handle): https://hdl.handle.net/20.500.11752/OPEN-2123
Repository name	CLARIN-IT Repository (ILC-CNR, Institute for Computational Linguistics "A. Zampolli"), hosted within the CLARIN infrastructure.
Object name	`KIP-transcripts-v1.1.0.zip`
Format names and versions	ZIP archive containing: `.eaf` (ELAN time-aligned XML transcription files); `.txt` (linearized Jefferson-style transcripts); `.txt` (linearized orthographic transcripts); `.tsv` (tokenized tab-separated files); metadata files and documentation (`README.md`, `CITATION.cff`, `LICENSE`). Version: v1.1.0.
Creation dates	Data collection: 2016-01-01 to 2019-12-31 (recordings conducted between 2016 and 2019).
Dataset creators	Silvia Ballarè (University of Bologna) – data revision and pseudonymization; Eugenio Goria (University of Turin) – corpus preparation, module coordination; Caterina Mauri (University of Bologna) – scientific coordination and project supervision.
Language	Italian (standard Italian, with occasional regional varieties); English (used for documentation).
License	Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY-NC-SA 4.0).
Publication date	2019-06-04.
ParlaTO transcripts	
Repository location	DOI: https://doi.org/10.60760/unibo/parlato Persistent identifier (Handle): https://hdl.handle.net/20.500.11752/OPEN-2125
Repository name	CLARIN-IT Repository (ILC-CNR, Institute for Computational Linguistics "A. Zampolli"), within the CLARIN infrastructure.
Object name	`ParlaTO-transcripts-v1.1.0.zip`
Format names and versions	ZIP archive containing: `.eaf` (ELAN time-aligned XML transcription files); `.txt` (linearized Jefferson-style transcripts); `.txt` (linearized orthographic transcripts); `.tsv` (tokenized tab-separated files); metadata and documentation files. Version: v1.1.0.

(Contd.)

Creation dates	Data collection: 2018-01-01 to 2020-12-31 (semi-structured interviews conducted between 2018 and 2020).
Dataset creators	Silvia Ballarè (University of Bologna) – module coordination and corpus preparation; Massimo Cerruti (University of Turin) – scientific coordination and project supervision.
Language	Italian (spoken Italian, with regional variation typical of Turin and Piedmont); English (used for documentation).
License	Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY-NC-SA 4.0).
Publication date	2020-06-01.
KIPasti transcripts	
Repository location	DOI: https://doi.org/10.60760/unibo/kipasti Persistent identifier (Handle): https://hdl.handle.net/20.500.11752/OPEN-2124
Repository name	CLARIN-IT Repository (ILC-CNR, Institute for Computational Linguistics "A. Zampolli"), within the CLARIN infrastructure.
Object name	`KIPasti-transcript-v1.1.0.zip`
Format names and versions	ZIP archive containing: `.eaf` (ELAN time-aligned XML transcription files); `.txt` (linearized Jefferson-style transcripts); `.txt` (linearized orthographic transcripts); `.tsv` (tokenized tab-separated files); metadata and documentation files. Version: v1.1.0.
Creation dates	Data collection: 2020-01-01 to 2024-12-31 (mealtime conversations recorded between 2020 and 2024).
Dataset creators	Silvia Ballarè (University of Bologna) – module coordination and corpus design; Caterina Mauri (University of Bologna) – module coordination and corpus design; Eleonora Zucchini (University of Bologna) – data revision and pseudonymization.
Language	Italian (predominantly), with frequent occurrences of regional dialect varieties; English (used for documentation).
License	Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY-NC-SA 4.0).
Publication date	2024-04-30.
ParlaBO transcripts	
Repository location	DOI: https://doi.org/10.60760/unibo/parlabo Persistent identifier (Handle): https://hdl.handle.net/20.500.11752/OPEN-2126
Repository name	CLARIN-IT Repository (ILC-CNR, Institute for Computational Linguistics "A. Zampolli"), within the CLARIN infrastructure.
Object name	`ParlaBO-transcripts-v1.1.0.zip`
Format names and versions	ZIP archive containing: `.eaf` (ELAN time-aligned XML transcription files); `.txt` (linearized Jefferson-style transcripts); `.txt` (linearized orthographic transcripts); `.tsv` (tokenized tab-separated files); metadata and documentation files (`README.md`, `CITATION.cff`, `LICENSE`). Version: v1.1.0.
Creation dates	Data collection: 2021-01-01 to 2024-12-31 (semi-structured interviews conducted between 2021 and 2024 in Bologna and its province).
Dataset creators	Silvia Ballarè (University of Bologna) – module coordination and corpus preparation; Caterina Mauri (University of Bologna) – module coordination and corpus preparation; Eleonora Zucchini (University of Bologna) – data revision and pseudonymization.
Language	Italian (spoken Italian, with regional variation typical of Bologna and Emilia-Romagna, including occasional dialectal segments); English (used for documentation).
License	Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY-NC-SA 4.0).
Publication date	2024-10-01.
StraParlaBO and StraParlaTO transcripts	
–	currently being deposited.

3 REUSE PROCESS AND METHODOLOGY

KIParla is organized into modules, each consisting of speech events. Within ELAN, each speech event is structured into tiers corresponding to individual speakers. Speech is segmented into "transcription units" (TUs), anchored to specific offsets in the transcription file and primarily as practical working tools rather than as theoretically defined intonational units.

Table 2 Transcription conventions.

SYMBOL	MEANING	TYPE	LEVEL
`word,`	Weakly rising intonation	Categorical	Token
`word?`	Rising intonation	Categorical	Token
`word.`	Falling intonation	Categorical	Token
`wo:rd`	Prolonged sound	Categorical	Character
`(.)`	Short pause	Categorical	Token
`word=word`	Prosodically linked units		
`°word°`	Lower volume	Categorical	Token
`WORD`	Higher volume	Categorical	Token
`wor-`	Interrupted word	Categorical	Token
`>word<`	Faster speech	Categorical	Span
`<word>`	Slower speech	Categorical	Span
`[word]`	Overlapping speech	Relational	Span
`(word)`	Uncertain transcription (transcriber's guess)	Categorical	Span
`xxx`	Unintelligible sequence (one x per syllable)	Categorical	Token
`((laughs))`	Non-verbal behavior	Categorical	Token

Transcriptions are produced by trained human annotators who apply Jefferson conventions (Table 2) competently and consistently, but without necessarily reflecting on the structural status of the elements they produce. They transcribe words, pauses, overlaps, and prosodic phenomena in accordance with established practice; however, the corpus itself does not explicitly encode what its stable, computationally tractable units of storage and maintenance are. The validation pipeline we implement makes the analytical units implicit in Jefferson transcription operational and inspectable.[1]

Although for qualitative analysis Jeffersonian flexibility might be a strength, to make transcripts fully machine-checkable, it requires an explicit operationalization. A first step of our pipeline therefore concerned the precise definition of explicit analytical units and then to provide validation criteria relative to those declared units.

Each symbol reported in Table 2 can be characterized along two orthogonal dimensions: (i) the type of information it encodes (categorical vs. relational), and (ii) the level at which it is anchored (character-based, token-based, span-based, or unit-based). By systematically classifying symbols along these dimensions, the pipeline derives three explicit representational objects (TUs, spans, and tokens) and enforces well-formedness constraints at each level.

3.1 TRANSCRIPTION UNITS: STRUCTURAL NORMALIZATION AND COHERENCE

The conversion process begins by treating the transcript as an ordered sequence of speaker-attributed TUs, operationally defined adspeaker-labeled segments associated with a time interval and a Jefferson-formatted annotation string. The workflow accepts segmentation as given, but subjects it to structural validation.

1 The full pipeline — including the validation scripts, the regex-based well-formedness checks, the overlap resolution algorithm, and the serialization routines — is publicly available at https://github.com/KIParla/tools. The repository is currently under active development; a citable, versioned release will be archived with a persistent identifier upon completion of the ongoing consolidation work.

DOI: 10.5334/johd.527

Each TU is standardized to remove extraneous symbols, regularize spacing and normalize formatting irregularities: this ensures that surface variability does not propagate into downstream structural analysis.

Validation also verifies that each TU is structurally well-formed: it must be anchored to a coherent time interval (`end > start`), and must contain interpretable annotation content. Units that collapse to empty or purely non-linguistic material after normalization can be excluded from further processing. The result of this stage is a temporally ordered sequence of normalized TUs that are internally consistent and suitable for more fine-grained structural analysis.

Each unit is finally assigned an ID unique within the speech event.

3.2 SPANS: DELIMITER BALANCE AND ALIGNMENT WITH TEMPORAL OVERLAP

The second stage concerns span-like phenomena, that is, Jefferson devices that extend over a contiguous stretch of text. These include low-volume segments, altered pace segments, guessing/uncertain hearing markers, and bracketed overlap regions.

Span-level validation proceeds in two steps. First, delimiter balance is verified within each TU. All paired markers must be properly opened and closed; unbalanced or malformed delimiters are treated as structural errors. This ensures that the annotation string can be interpreted as a coherent sequence of span delimiters that are locally balanced, without imposing a strict nesting constraint across different Jefferson devices: in keeping with Jeffersonian practice, overlapping or interleaving patterns across distinct marker types (e.g., *la seconda lingua strani[era <in]glese:> cioè >la la prima lingua [stran]iera inglese< oppure,* 'my second foreign language english well my my first foreign language english or') are treated as formally admissible, provided that each delimiter type is internally well-formed.

This interpretation also requires disambiguating symbols that can function both as opening and closing delimiters, such as the angle bracket. In Jefferson notation, the same graphic symbol (e.g., `>`) may mark either the beginning or the end of a paced segment, depending on context. To render such cases computationally tractable, we adopt explicit structural assumptions: for instance, stretches marked by `>…<` and `<…>` are treated as internally coherent units that cannot be interwoven or ambiguously concatenated.

Spans must be anchored to a clearly defined address space: in this framework, spans are linked to token indices. Validation ensures that span boundaries refer to existing tokens (i.e., cover at least one linguistic character) and are non-empty. To this end, certain surface configurations are normalized prior to span projection. For example, an opening parenthesis cannot be followed by a space (e.g., `[ word ↦ [word`), and a closing parenthesis cannot be preceded by a space, so that the delimited material is directly attached to the relevant token rather than floating between tokens. Similarly, in the case of vowel prolongations, spans must include at least one alphabetic character of the affected token. Configurations in which the span would otherwise consist solely of prolongation markers are therefore adjusted: sequences such as `word[:: ↦ wor[d::`, and sequences such as `word]. ↦ word.]`, ensuring that the span includes at least one segment of lexical material and can be reliably mapped onto a token-level character range. Overlaps require special attention as they are validated against the temporal structure of the transcript.

They represent the most complex phenomenon in the conversion pipeline, because their identification relies on two partially independent sources of information: (i) temporal alignment, and (ii) textual span annotation using Jefferson brackets. The working assumption of the pipeline is that these two sources should converge: overlap events detectable from time intervals should correspond to bracketed overlap spans in the transcription string. The algorithm therefore derives overlap events from temporal data and subsequently matches them to annotated spans.

The first step consists in constructing a temporal overlap graph over TUs. Each unit is represented as a node; an undirected edge is introduced between two nodes if their time intervals intersect (i.e., if both units are simultaneously active for a non-zero duration). Since a unit may overlap with multiple other units the resulting structure is generally not a simple

pairwise mapping but a graph encoding all temporal intersections. Figures 4 and 5 illustrate two prototypical configurations that motivate the graph-based treatment of overlap. In the first case, a single TU participates in more than one temporally distinct overlap event: one unit overlaps with a second unit during an initial time window and later overlaps with a third unit during a different time window. This situation involves three units overall, it corresponds to two separate overlap events, because there is no time interval during which all three units are simultaneously active. In the overlap graph, this configuration yields two distinct cliques sharing a common node. In the second case, three speakers are simultaneously active within a shared temporal interval. Here, the overlap graph contains a fully connected subgraph among the three corresponding units, forming a maximal clique. This clique defines one overlap event whose temporal boundaries are given by the maximum of the units' start times and the minimum of their end times.

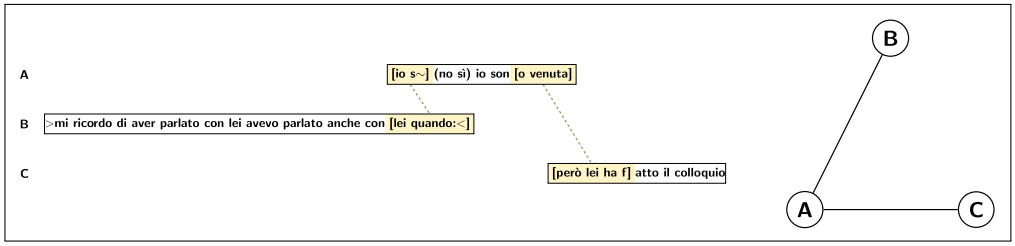

Figure 4 Depiction of a case where one TU (A) participates in two distinct overlap events. Example extracted from Figure 2.

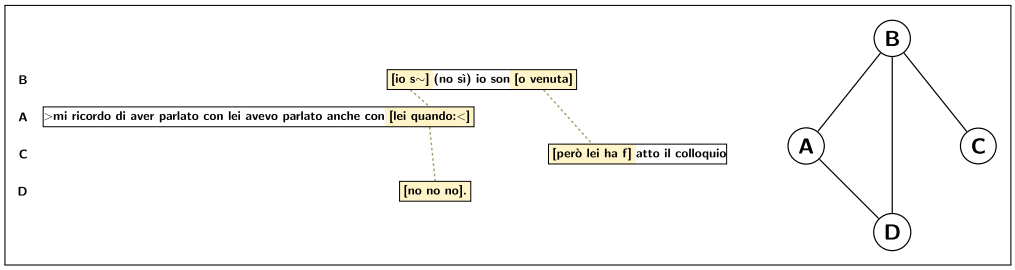

Figure 5 Depiction of a case where one TU (B) participates in two distinct overlap events. Example extracted from Figure 2.

Overlap is therefore not defined pairwise but event-wise, as maximal cliques (a set of two or more units that are mutually overlapping during a shared temporal window and that cannot be extended to more units) on the graph. For each clique, the algorithm computes its temporal boundaries that defines the effective time span during which all units in the clique are simultaneously active. Each such interval is treated as a distinct overlap event and assigned a unique identifier within the speech event. Overlap events can then be ordered chronologically based on their onset time.

The validator also performs minor refinements shifting unit boundaries when time-derived overlaps seem spurious.

For each TU, the algorithm determines which overlap events include that unit. Ideally, if a unit participates in n temporal overlap events, the Jefferson transcription of that unit should contain n bracketed overlap spans. To establish correspondence, both temporal events and textual spans are ordered, based on their onset time or their position in the annotation string. The matching then proceeds sequentially.

The matching procedure is subject to additional constraints such as excluding nonverbal tokens from overlap participation and ignoring spans that violate structural constraints.

Each token that falls within a textual overlap span inherits the identifier of the corresponding temporal overlap event. In this way, overlap becomes a relational structure linking tokens across units. Concretely, if tokens t_1 (unit A), t_2 (unit B), and t_3 (unit D) in Figure 5 all belong to the same overlap event identifier, the vertical format makes explicit that these tokens were produced within the same temporal overlap window. This makes explicit information that is only implicit in Jefferson transcription. Human annotators mark the presence of overlap locally, but they do not specify which bracket in one unit corresponds to which bracket in another.

3.3 TOKENS: LOCAL WELL-FORMEDNESS AND FEATURE EXTRACTION

The final stage operates at the level of tokens. Because Jefferson conventions do not define token boundaries, the workflow adopts an explicit tokenization scheme. Tokenization proceeds within each TU by splitting the normalized annotation on whitespace while also treating the prosodic-latching marker (=) as a separable element; the splitter preserves = as its own item so that it can be interpreted structurally rather than as part of a wordform. In addition, when an apostrophe separates alphabetic material on both sides, the implementation splits the string into two subtokens and marks the first as having no following space, capturing the orthographic attachment relevant for Italian clitics and elisions while maintaining a consistent notion of token boundaries.

Token-level validation enforces local well-formedness constraints. Each token must conform to the permitted character inventory and to the expected surface patterns for lexical forms or recognized Jefferson markers.

```
(\pL:*)+[-']?[.,?]?
```

Tokens are then assigned to a category: short pauses, nonverbal behaviours, unintelligible tokens and lexical items are the main labels. Multiword non-verbal behaviors are concatenated in our pipeline using an underscore (e.g., ((risata_lunga))) in order to prevent internal tokenization and to retain them as single tokens in the final representation.

Prosodic information encoded on token characters is systematically extracted and represented as structured features. Final punctuation (i.e., `.`, `,` and `?`) determines intonational contour categories; interruption symbols (`'` or `-`) signal truncation; colon sequences encode vowel prolongation; capitalization is interpreted as a marker of increased volume and normalized in the lexical representation. In each case, the surface marker is separated from the underlying lexical form while preserving its interpretive value as a feature of the token.

After tokenization, span information is projected onto tokens. Character-level spans identified at the previous stage are mapped onto the tokens they cover, and the relevant token-internal character ranges are recorded. This allows, for example, for a low-volume or fast-paced span to be associated precisely with the lexical material it affects, even when spans cut across token boundaries.

The result of token-level validation is a sequence of tokens that are locally well-formed, categorically typed, and enriched with prosodic and interactional features derived from the Jefferson surface notation.

Each token was assigned a unique identifier derived from the TU and its position within it, ensuring traceability back to the original source.

3.4 INTEGRATING THE THREE LEVELS

The three layers provide an explicit operationalization of Jefferson-style transcription. Without attempting to "validate Jefferson" as a whole, the pipeline validates a declared analytical model: TUs are time-aligned speaker segments with normalized annotation strings; spans are balanced and extractable realizations of selected Jefferson devices; and tokens are locally well-formed units combining lexical identity with structured prosodic features. This layered design preserves the descriptive richness of Jefferson notation while enabling reproducible conversion. A central robustness criterion is round-trip stability: tokenization and serialization must reproduce a canonical version of the transcript (modulo formatting normalization). This constraint ensures that the induced grammar is internally coherent and that no information is lost or ambiguously represented. In the vertical format (Figure 6), each token occupies a single row enriched with speaker, structural, alignment, and linguistic attributes (the full specification is given in Table 3), making it possible to query lexical, interactional, and metadata dimensions jointly: capabilities not available in the original transcription-based representation.

All span-like features and character-based phenomena are represented using explicit character offsets relative to the normalized orthographic form of each token. Temporal alignment from ELAN is preserved without redundancy: timestamps are anchored to the first and last token of each TU, allowing deterministic reconstruction of time-aligned units while keeping alignment conceptually tied to the unit rather than individual words. Interactional

and prosodic phenomena that apply to subtoken segments are encoded through explicit character offsets relative to the normalized token form. This character-level anchoring maintains fine-grained correspondence with the original Jefferson string and guarantees faithful round-trip reconstruction.

COLUMN	DESCRIPTION
`token_id`	Unique identifier of the token within the conversation. It consists of two integers separated by a hyphen (`^\d+-\d+$`). The first encodes the ID of the TU. It must be treated as an opaque identifier.
`speaker`	Either a valid KIParla speaker code or `???` to signal missing participant metadata (`!^[A-Z]+[0-9]+|\?3$ !`). Parsed as a categorical string.
`tu_id`	Progressive identifier of the TU (`^\d+`). Tokens sharing the same `tu_id` form a contiguous block and reconstruct one validated TU.
`span`	Exact substring of the original Jefferson transcription corresponding to the token, including all symbols (e.g., colons, brackets, punctuation, variation markers). Must be treated as a literal string without normalization.
`form`	Normalized orthographic representation of the token. Jefferson-specific symbols are removed and encoded in dedicated feature columns. Short pauses are encoded as `[PAUSE]`; unintelligible tokens are normalized to `x`. Parsed as a Unicode string representing the orthographic base form.
`type`	Categorical label describing the token class. Possible values include `linguistic`, `nonverbalbehavior`, `shortpause`, `unknown`, and `error` (reserved for cases requiring manual inspection). Parsed as a closed-set categorical variable.
`jefferson_feats`	Pipe-separated list of key–value pairs in the format `Feature=Value`. The field may be empty. Parsers should split on `|`, then on `=`. Possible values are described in Table 4. Field may be empty.
`align`	Alignment metadata, present only on the first and last token of each TU. It is encoded as a pipe-separated key–value pairs: `AlignBegin=<float>` and/or `AlignEnd=<float>`, expressed in seconds. Enables deterministic reconstruction of time-aligned units.
`prolongations`	Comma-separated list of elongation encodings in the format `<char_id>x<count>`. `char_id` is the zero-based index in `form`; `count` is the number of consecutive colons in the original `span`. Example: `2x2,6x1`. Field may be empty.
`pace`	Encodes participation in fast or slow spans. Either empty or one of `Fast=<start>-<end>` or `Slow=<start>-<end>`, where indices are zero-based over `form`.
`guesses`	Comma-separated list of uncertain character ranges in the format `<start>-<end>`, using zero-based indices over `form`. Field may be empty.
`overlaps`	Comma-separated list of overlap encodings in the format `<start>-<end>(<overlap_id>)`. Indices are zero-based over `form`; `overlap_id` is the progressive identifier of the temporal overlap event derived via the clique-based algorithm. If unresolved, the identifier is `?`.

Table 3 Description of the columns composing the TSV format. The file contains a header useful to interpret columns.

FEATURE	VALUES
`SpaceAfter`	`No` when tokenization happens on apostrophe
`ProsodicLink`	`Yes` when tokenization happens on the `=` symbol
`Intonation`	`Falling, Rising, WeaklyRising` for `. , ?` respectively
`Interrupted`	`Yes` for tokens ending in `–`
`Truncated`	`Yes` for tokens ending in `'`
`Volume`	`High, Low` for tokens in capitalized spans or spans enclosed in `°...°`

Table 4 Feature-Value pairs for column jefferson_feats.

Over time, this verticalized representation has evolved from an intermediate export into the authoritative pivot format of the corpus. Instead of treating ELAN files as the primary artefact and generating derivative formats ad hoc, the workflow now centers on the pivot

DOI: 10.5334/johd.527

1	28-0	BO026	28	>mi	mi	linguistic	_	Begin=50.761	_	Fast=0-2	_	_
2	28-1	BO026	28	ricordo	ricordo	linguistic	_	_	_	Fast=0-7	_	_
3	28-2	BO026	28	di	di	linguistic	_	_	_	Fast=0-2	_	_
4	28-3	BO026	28	aver	aver	linguistic	_	_	_	Fast=0-4	_	_
5	28-4	BO026	28	parlato	parlato	linguistic	_	_	_	Fast=0-7	_	_
6	28-5	BO026	28	con	con	linguistic	_	_	_	Fast=0-3	_	_
7	28-6	BO026	28	lei	lei	linguistic	_	_	_	Fast=0-3	_	_
8	28-7	BO026	28	avevo	avevo	linguistic	_	_	_	Fast=0-5	_	_
9	28-8	BO026	28	parlato	parlato	linguistic	_	_	_	Fast=0-7	_	_
10	28-9	BO026	28	anche	anche	linguistic	_	_	_	Fast=0-5	_	_
11	28-10	BO026	28	con	con	linguistic	_	_	_	Fast=0-3	_	_
12	28-11	BO026	28	[lei	lei	linguistic	_	_	_	Fast=0-3	_	0-3(245)
13	28-12	BO026	28	quando:<]	quando	linguistic	_	End=53.754	5x1	Fast=0-6	_	0-6(245)
14	29-0	BO039	29	[no	no	linguistic	_	Begin=52.815	_	_	_	0-2(245)
15	29-1	BO039	29	no	no	linguistic	_	_	_	_	_	0-2(245)
16	29-2	BO039	29	no].	no	linguistic	Intonation=Falling	End=53.662	_	_	_	0-2(245)
17	30-0	BO038	30	[io	io	linguistic	_	Begin=53.465	_	_	_	0-2(245)
18	30-1	BO038	30	s~]	s~	linguistic	Interrupted=Yes	_	_	_	_	0-2(245)
19	30-2	BO038	30	(no	no	linguistic	_	_	_	_	0-2(0)	_
20	30-3	BO038	30	sì)	sì	linguistic	_	_	_	_	0-2(0)	_
21	30-4	BO038	30	io	io	linguistic	_	_	_	_	_	_
22	30-5	BO038	30	son[o	sono	linguistic	_	_	_	_	_	3-4(8)
23	30-6	BO038	30	venuta::]	venuta	linguistic	_	End=54.858	5x2	_	_	0-6(8)
24	31-0	BO026	31	[però	però	linguistic	_	Begin=54.24	_	_	_	0-4(8)
25	31-1	BO026	31	lei	lei	linguistic	_	_	_	_	_	0-3(8)
26	31-2	BO026	31	ha	ha	linguistic	_	_	_	_	_	0-2(8)
27	31-3	BO026	31	f]atto	fatto	linguistic	_	_	_	_	_	0-1(8)
28	31-4	BO026	31	il	il	linguistic	_	_	_	_	_	_
29	31-5	BO026	31	colloquio,	colloquio	linguistic	Intonation=WeaklyRising	End=55.748	_	_	_	_
30	32-0	BO039	32	e:h	eh	linguistic	_	Begin=55.833	0x1	_	_	_
31	32-1	BO039	32	con	con	linguistic	_	_	_	_	_	_
32	32-2	BO039	32	la	la	linguistic	_	_	_	_	_	_
33	32-3	BO039	32	ro[ssi]	rossi	linguistic	_	_	_	_	_	2-5(9)
34	32-4	BO039	32	l'	l'	linguistic	SpaceAfter=No	_	_	_	_	_
35	32-5	BO039	32	ho	ho	linguistic	_	_	_	_	_	_
36	32-6	BO039	32	fatto.	fatto	linguistic	Intonation=Falling	_	_	_	_	_
37	32-7	BO039	32	[sì].	sì	linguistic	Intonation=Falling	End=57.45	_	_	_	0-2(10)
38	33-0	BO026	33	[sì]	sì	linguistic	_	Begin=56.654|End=56.773	_	_	_	0-2(9)
39	34-0	BO026	34	[sì]	sì	linguistic	_	Begin=57.177	_	_	_	0-2(10)
40	34-1	BO026	34	sì.	sì	linguistic	Intonation=Falling	End=57.668	_	_	_	_
41	35-0	BO026	35	anche	anche	linguistic	_	Begin=58.027	_	_	_	_
42	35-1	BO026	35	lei?	lei	linguistic	Intonation=Rising	End=58.552	_	_	_	_
43	36-0	BO038	36	sì.	sì	linguistic	Intonation=Falling	Begin=58.64	_	_	_	_
44	36-1	BO038	36	(.)	[PAUSE]	shortpause	_	_	_	_	_	_
45	36-2	BO038	36	sì	sì	linguistic	_	_	_	_	_	_
46	36-3	BO038	36	per~	per~	linguistic	Interrupted=Yes	_	_	_	_	_
47	36-4	BO038	36	cioè	cioè	linguistic	_	_	_	_	_	_
48	36-5	BO038	36	io	io	linguistic	_	_	_	_	_	_
49	36-6	BO038	36	sono	sono	linguistic	_	_	_	_	_	_
50	36-7	BO038	36	venuta	venuta	linguistic	_	_	_	_	_	_
51	36-8	BO038	36	anche:	anche	linguistic	_	_	4x1	_	_	_
52	36-9	BO038	36	[a	a	linguistic	_	_	_	_	_	0-1(11)
53	36-10	BO038	36	ricevimento	ricevimento	linguistic	_	_	_	_	_	0-11(11)
54	36-11	BO038	36	con	con	linguistic	_	_	_	_	_	0-3(11)
55	36-12	BO038	36	lei].	lei	linguistic	Intonation=Falling	_	_	_	_	0-3(11)
56	36-13	BO038	36	sì	sì	linguistic	_	_	_	_	_	_
57	36-14	BO038	36	sì	sì	linguistic	_	_	_	_	_	_

Figure 6 Verticalized format of conversation BOA1010.

representation, from which ELAN files and other task-specific formats can be regenerated reproducibly (Figure 7). Crucially, the architecture is bidirectional: updates or corrections applied at the pivot level propagate to derived formats, and new annotation layers can be aligned back to the core representation. This enables updates, corrections, or enrichments introduced at one level to propagate across representations, reducing divergence and duplication. This projection-based approach mitigates a common problem in corpus development, namely the proliferation of partially incompatible versions of the same data.

From a maintenance perspective, centralizing development around the verticalized format has led to tangible benefits. Versioning, validation, and consistency checks can be performed at a single point, simplifying long-term corpus management. This has proven particularly valuable in the context of a modular corpus such as KIParla, where new data are periodically added and existing modules may require revision.

4 OUTCOMES AND EXPERIENCE

The introduction of the verticalized pivot format has had immediate methodological and infrastructural consequences. Most notably, it has enabled large-scale lemmatization and part-of-speech tagging of the entire corpus. Extracting stable token sequences directly from tier-based ELAN annotations would have required substantial preprocessing and would have made reintegration of morphosyntactic information into the transcription layer extremely difficult. By contrast, the TSV format makes tokens explicit and structurally independent of alignment metadata, allowing standard NLP pipelines to be applied directly while preserving traceability to the original transcription (Figure 8).

Tokens of type `linguistic` and `unknown` were processed through UDPipe Straka (2018) for UD-style tokenization and morphological analysis, while `shortpause` and `nonverbalbehavior` tokens were excluded from syntactic processing. Because UD tokenization may introduce multiword expansions that do not correspond one-to-one with transcription tokens, newly created sub-tokens were inserted into the vertical representation using extensions of the original token_id (e.g., 254-1a, 254-1b) and marked as type `syntactic`. Two additional columns, `lemma` and `upos`, store the resulting annotations. This design preserves

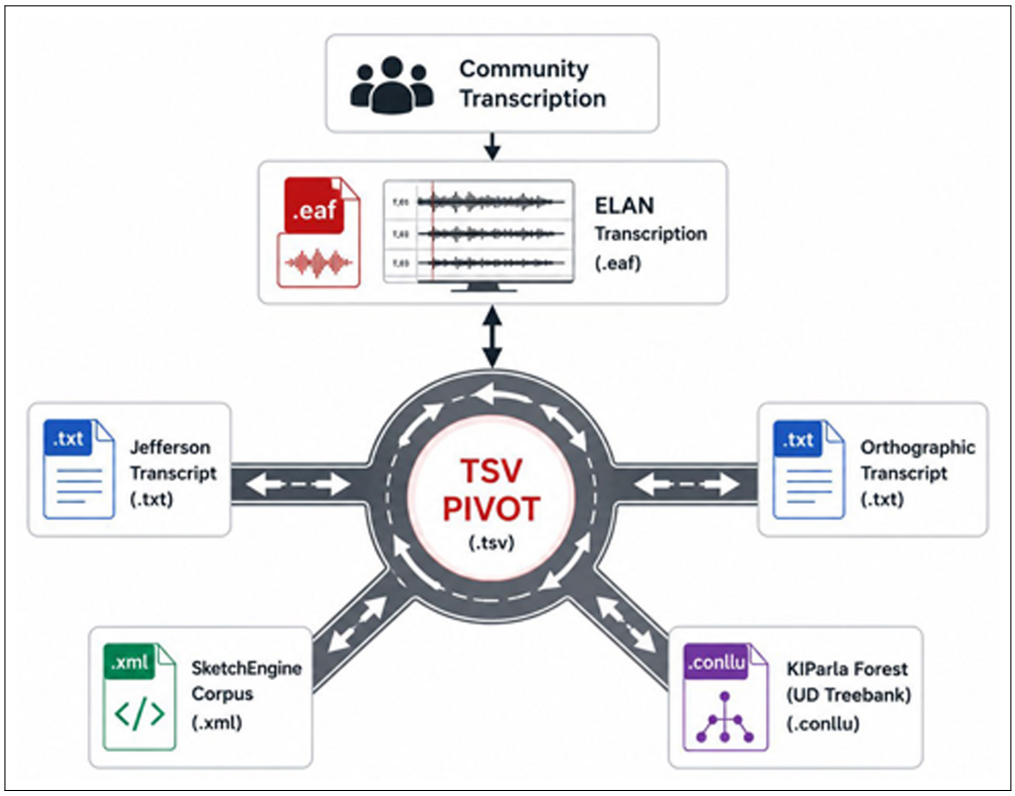

Figure 7 Towards a new bidirectional pipeline.

1	token_id	speaker	tu_id	unit	id	span	form	lemma	upos	jefferson_feats
2	0-0	PKP025	0	0	1	faccio	faccio	fare	VERB	_
3	0-1	PKP025	0	0	2	il	il	il	DET	_
4	0-2	PKP025	0	0	3	tè	tè	tè	NOUN	_
5	1-0	PKP026	1	1	1	(°allora°)	allora	allora	ADV	_
6	2-0	PKP025	2	2	1	[accendo]?	accendo	accendere	VERB	_
7	3-0	PKP026	3	3	1	[va	va	andare	VERB	_
8	3-1	PKP026	3	3	2	bene]	bene	bene	ADV	_
9	4-0	PKP026	4	4	1	ci	ci	ci	PRON	_
10	4-1	PKP026	4	4	2	fermiamo	fermiamo	fermare	VERB	_
11	4-2	PKP026	4	4	3	un	un	uno	DET	_
12	4-3	PKP026	4	4	4	attimo	attimo	attimo	NOUN	_

Figure 8 Conversation KPC001 after lemmatization and POS tagging. Jefferson features are not shown for reasons of space.

the integrity of the transcription layer while enabling UD-compatible syntactic annotation in a fully aligned and reversible manner.

A concrete example is provided by Italian articulated prepositions such as *del* (*di* + *il*), which UD tokenizes as two distinct syntactic units. In the pivot representation, the surface transcription token *del* at position `1-1` is expanded into two sub-token rows `1-1a` (*di*, ADP) and `1-1b` (*il*, DET), both carrying `type=syntactic`. Crucially, because syntactic sub-tokens are identified by a dedicated type value, the entire syntactic expansion layer can be removed from the representation with a single standard command (`grep -v syntactic`), yielding a clean transcription-level view without any UD expansion. This design ensures that the syntactic layer is both tightly integrated and trivially isolatable, with no structural consequences for the underlying interactional representation.

The vertical architecture also supports efficient and auditable manual revision: corrections to lemmas, tags, or even transcription errors can be localized to specific rows in the TSV. Once updated in the pivot representation, changes propagate deterministically to all derived artifacts. This enables a DevOps-like curation workflow in which modifications are incremental,

traceable, and version-controlled, avoiding opaque edits to monolithic transcription files (cf. Steiner 2017).

A further outcome of this restructuring is the derivation of a syntactic treebank from KIParla Pannitto et al. (2025). The structural proximity of the pivot format to CoNLL-U supports direct projection of dependency annotations while preserving interactional information. Crucially, the pivot representation allows a principled distinction between phenomena that should become syntactic tokens and those that should remain token-level metadata. For example, short pauses are encoded in the CoNLL-U MISC field (e.g., `PauseAfter=Yes`), and overlap information, already resolved at token level, is projected as explicit cross-references between overlapping tokens. This yields a treebank in which syntactic structure remains interoperable with prosodic and interactional features, enabling integrated cross-layer queries. KIParla Forest (Pannitto et al., 2025) was first introduced in UD v2.17 and an update was released in UD v2.18. The treebank currently hosts 6 conversations from the original corpus, an interview (BOD2018, comprising 2 speakers), a free conversation (BOA3017, comprising 4 speakers), three office-hours conversations (BOA1003, BOA1008, BOA1009, comprising two speakers each), and a university lecture (TOD1005bis, comprising one speaker) for a total of 18, 050 tokens and 2, 221 sentences.

Spoken interaction also raises well-known challenges for sentence segmentation. Rather than imposing a fixed solution, the pivot format isolates segmentation as an explicit and revisable layer. TUs provide a conservative default boundary for UD processing, but alternative segmentation schemes can be introduced through additional columns without modifying the underlying token structure. In this way, multiple segmentation strategies can coexist within the same representation.

More generally, the column-based architecture of the pivot format ensures extensibility. New annotation layers can be added as columns without restructuring existing data, preserving backward compatibility while enabling further enrichment (e.g., discourse, information-structural, or pragmatic annotation). The pivot format thus functions not merely as a conversion layer, but as an enabling infrastructure: it retains the descriptive richness of Jefferson-style transcription while making the corpus computationally tractable, interoperable with established standards, and incrementally extensible.

5 RECOMMENDATIONS AND GOOD PRACTICES

We restructured the KIParla corpus around a vertical, pseudo-tokenized pivot format that reconciles the Jefferson-style transcription's richness with the technical needs of scalable maintenance, validation, and computational reuse (much in the spirit of Steiner 2017). From a corpus initially optimized for human-readable interactional analysis, we added an explicit preservation layer that formalizes the analytical units implicit in the transcription and enforces well-formedness and machine-readability at every level.

The TSV format resulting from validation functions as a pivot between interaction-oriented annotation and computational infrastructures. Each token is represented as a stable record enriched with transcription-derived features, preserving invertibility to the original transcription. The format is intentionally designed to support incremental correction and version-controlled maintenance: revisions are localized at the level of token identifiers and propagate deterministically to all derived formats.

This restructuring further enables the derivation of new projects, such as a syntactic treebank fully interoperable with the interactional layer.

More broadly, the proposed approach demonstrates that spoken-language corpora need not choose between human-readable richness and computational rigor. By introducing an explicit preservation unit and enforcing layered validation, it is possible to maintain Jefferson-style descriptive fidelity while enabling scalable, DevOps-inspired curation workflows based on automation, reproducibility, and interoperability. The KIParla pivot format thus serves not only as a technical solution for a growing corpus of Italian spoken interaction, but also as a model for sustainable spoken-language data engineering in contemporary research infrastructures. Potentially, any input format containing information that can be made explicit in a vertical representation can be aligned to this pivot-format.

The adoption of a pivot format has made versioning a central concern rather than an afterthought: version control systems allow curators to record when and why modifications were introduced, to compare alternative solutions, and to revert changes when necessary. Recent work in the linguistic annotation community has begun to articulate similar proposals (Waldon & Schneider, 2025).

Versioning should apply to all data artefacts, not just scripts or software. Treating datasets as versioned objects aligns corpus curation with software development practices and supports transparent reuse, especially in collaborative, long-term projects. Corpora are thus living artefacts, not static publications; their evolution must be traceable, auditable, and reversible.

The KIParla experience suggests that dataset curation increasingly resembles a form of continuous development, rather than a linear production pipeline culminating in a static release. New data are added, errors are corrected, formats evolve, and new reuse scenarios emerge over time. This DevOps-inspired approach to language resource engineering lays a foundation for sustainable corpus infrastructure, keeping room for expert interventions and allowing the resource to seamlessly integrate into new research contexts.

From our perspective, interoperability is not achieved through adherence to formats and standards, but through the alignment of curatorial practices, particularly in the case of spoken language resources.

Several developments are already underway to consolidate and extend the infrastructure described in this paper.

DATA ACCESSIBILITY STATEMENT

The KIParla datasets described in this paper are deposited in the CLARIN-IT Repository (ILC-CNR), part of the European CLARIN Research Infrastructure, and are available under the Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International (CC BY-NC-SA 4.0) license.

The datasets can be accessed through the following persistent identifiers:

- KIP: DOI https://doi.org/10.60760/unibo/kip; Handle https://hdl.handle.net/20.500.11752/OPEN-2123
- ParlaTO: DOI https://doi.org/10.60760/unibo/parlato; Handle https://hdl.handle.net/20.500.11752/OPEN-2125
- KIPasti: DOI https://doi.org/10.60760/unibo/kipasti; Handle https://hdl.handle.net/20.500.11752/OPEN-2124
- ParlaBO: DOI https://doi.org/10.60760/unibo/parlabo; Handle https://hdl.handle.net/20.500.11752/OPEN-2126

The validation, conversion, and maintenance tools used to generate and manage the KIParla pivot format are openly available at: https://github.com/KIParla/tools.

A versioned archival release of the software with a persistent identifier will be deposited upon completion of the ongoing consolidation work.

The derived Universal Dependencies treebank *KIParla Forest* is available through the Universal Dependencies repository: https://github.com/UniversalDependencies/UD_Italian-KIParlaForest.

ACKNOWLEDGEMENTS

We thank Eleonora Zucchini, Martina Simonotti, Silvia Ballarè, Bruno Guillaume, and members of the Task 1.5 of the UniDive COST Action CA21167, for various feedback on the development of the pivot format.

FUNDING STATEMENT

This research was developed within the PRIN 2022 PNRR Project “DiverSIta - Diversity in Spoken Italian” (PI: Caterina Mauri), funded by the European Union - NextGenerationEU with

the National Recovery and Resilience Plan (NRRP) - Mission 4 'Education and Research' - Component 2 'From Research to Business - Investment 1.1m Public Call PRIN 2022 PNRR published with the DDN. 1409 of 14.09/2022. MUR project code [P2022RFR8T] - CUP [J53D23017320001].

COMPETING INTERESTS

The authors have no competing interests to declare. Neither author is currently a member of the Journal of Open Humanities Data editorial team or board, nor has either held such a position within the last three years.

AUTHOR CONTRIBUTIONS

The authors meet the ICMJE criteria for authorship. Both authors contributed substantially to the conception and design of the work, participated in drafting and revising the manuscript, approved the final version, and agree to be accountable for all aspects of the work.

Ludovica Pannitto: Conceptualization, Methodology, Software, Formal Analysis, Data Curation, Validation, Visualization, Writing – Original Draft, Writing – Review & Editing.

Caterina Mauri: Conceptualization, Methodology, Project Administration, Funding Acquisition, Resources, Supervision, Validation, Writing – Original Draft, Writing – Review & Editing.

AUTHOR AFFILIATIONS

Ludovica Pannitto orcid.org/0000-0002-5760-6447
Department of Modern languages, literatures and cultures, University of Bologna, Bologna, Italy
Caterina Mauri orcid.org/0000-0002-8226-7846
Department of Modern languages, literatures and cultures, University of Bologna, Bologna, Italy

REFERENCES

Artstein, R., & **Poesio, M.** (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, *34*(4), 555–596. https://direct.mit.edu/coli/article/34/4/555-596/1999

Cerruti, M., & **Ballarè, S.** (2021). ParlaTO: corpus del parlato di torino. *Bollettino dell'Atlante Linguistico Italiano (BALI)*, *44*, 171–196.

Chrupała, G. (2023). Putting Natural in Natural Language Processing. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 7820–7827). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.495

Dobrovoljc, K. (2022). Spoken Language Treebanks in Universal Dependencies: an Overview. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 1798–1806). European Language Resources Association. https://aclanthology.org/2022.lrec-1.191/

Gagliardi, G. (2018). Inter-Annotator Agreement in linguistica: una rassegna critica. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)* (pp. 207–213). CEUR Workshop Proceedings. https://aclanthology.org/2018.clicit-1.37/

Goria, E., & **Mauri, C.** (2018). Il corpus KIParla: una nuova risorsa per lo studio dell'italiano parlato. In F. Masini & F. Tamburini (Eds.), *CLUB Working Papers in Linguistics* vol. 2 (pp. 96–116). Alma Mater Studiorum Università di Bologna.

Hedeland, H., & **Schmidt, T.** (2022). The TEI-based ISO Standard 'Transcription of spoken language'as an Exchange Format within CLARIN and beyond. *CLARIN Annual Conference* (pp. 34– 45). https://ecp.ep.liu.se/index.php/clarin/article/view/415

Institute of the Czech National Corpus (2025). NoSketch Engine. https://www.korpus.cz/noske. (Corpus query system).

Jefferson, G. (2004). Glossary of transcript symbols with an introduction. In G. H. Lerner (Ed.), *Pragmatics & Beyond New Series* vol. 125 (pp. 13–31). John Benjamins Publishing Company. https://benjamins.com/catalog/pbns.125.02jef

Linell, P. (2019). The Written Language Bias (WLB) in linguistics 40 years after. *Language Sciences*, *76*. https://doi.org/10.1016/j.langsci.2019.05.003

MacWhinney, B. (2019). *CHAT manual.* TalkBank. Retrieved 2026-02-24, from https://talkbank.org/0info/manuals/CHAT.pdf

Mauri, C., Ballarè, S., Goria, E., Cerruti, M., & **Suriano, F.** (2019). KIParla Corpus: A New Resource for Spoken Italian. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*.

Mauri, C., Ballarè, S., & **Zucchini, E.** (2024a). *Modulo KIPasti*. https://doi.org/10.60760/unibo/kipasti

Mauri, C., Ballarè, S., & **Zucchini, E.** (2024b). *Modulo ParlaBO*. https://doi.org/10.60760/unibo/parlabo

Mauri, C., Zucchini, E., Goria, E., & **Bernasconi, B.** (2025). *Il progetto DiverSIta. Documentare la diversità nell'italiano parlato*. CLUB – Circolo Linguistico dell'Università di Bologna. https://doi.org/10.6092/unibo/amsacta/8647

Max Planck Institute For Psycholinguistics, The Language Archive (2025). ELAN (version 7.0) [computer software]. https://archive.mpi.nl/tla/elan

Pannitto, L., Zucchini, E., Ballarè, S., Bosco, C., Mauri, C., & **Sanguinetti, M.** (2025). Introducing KIParla forest: seeds for a UD annotation of interactional syntax. In E. Hajičová & S. Kahane (Eds.), *Proceedings of the Eighth International Conference on Dependency Linguistics (depling, Syntaxfest 2025)* (pp. 54–73). Association for Computational Linguistics. https://aclanthology.org/2025.depling-1.5/

Steiner, I. (2017). A DevOps Manifesto for Speech Corpus Management. In *Proceedings of ESSV 2017*.

Straka, M. (2018). UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 197–207). Association for Computational Linguistics. https://www.aclweb.org/anthology/K18-2020

Waldon, B., & **Schneider, N.** (2025). A GitHub-based workflow for annotated resource development. In S. Peng & I. Rehbein (Eds.), *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)* (pp. 326–331). Association for Computational Linguistics. https://aclanthology.org/2025.law-1.27/

Werthmann, A. (2025). From spoken language data to TEI-based ISO. *Harmonizing language data: Standards for linguistic resources*, *4*, 145. https://doi.org/10.1515/9783112208212-007

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., & **Sloetjes, H.** (2006). ELAN: a professional framework for multimodality research. In *Proceedings of the fifth international conference on language resources and evaluation (LREC 2006)*. European Language Resources Association (ELRA). https://doi.org/10.63317/5pwa5zpssv4z

TO CITE THIS ARTICLE:
Pannitto, L., & Mauri, C. (2026). Reuse by Design: A Pivot-Based Architecture for the KIParla Corpus of Spoken Italian. *Journal of Open Humanities Data*, 12: 81. DOI: https://doi.org/10.5334/johd.527

Submitted: 01 March 2026
Accepted: 10 June 2026
Published: 24 Month 2026

Journal of Open Humanities Data is a peer-reviewed open access journal published by Ubiquity Press.