Complexity in Joint Corpus Representations, Under-specified Simultaneity, and the Case of KIParla

Martin Klotz and Thomas Krause and Anke Lüdeling
Humboldt-Universität zu Berlin
{martin.klotz,thomas.krause,anke.luedeling}@hu-berlin.de

Caterina Mauri and Ludovica Pannitto
Università di Bologna
{caterina.mauri,ludovica.pannitto}@unibo.it

Abstract

In this paper we describe joint corpus representations within a graph modeling framework of the corpus search engine ANNIS (Krause, 2019). With regard to their complexity, we distinguish three model classes, that are reflective of the underlying infrastructural and conceptual complexity of a corpus and provide an analysis on how to determine the model class for any corpus analytically from its design and build context. For integrating a corpus for register research, we derive a specific corpus model for the KIParla corpus (Mauri et al., 2019) and connect the complexity of its model to central aspects of its design and representation. The joint graph representation for KIParla is motivated to be of the highest complexity class and thus require the most general model known from parallel architectures.

1 Introduction

For corpus linguistic investigations, joint representations of all annotations enable efficient search and visualization of corpus data. A *joint representation* unifies all data in a corpus, making them accessible through a single query operation. From a conceptual and structural perspective, such representations are mathematical objects. From an implementation perspective, they are objects in computer memory. For illustration, some corpus search engines present their corpora to the user as a unified object, e. g. *ANNIS* (Krause, 2019), *Sketch Engine* (Kunilovskaya, Maria and Koviazina, Marina, 2017), *KorAP* (Diewald et al., 2016).[1] The general purpose of this paper is to identify different degrees or classes of complexity within such models. We derive and analytically motivate one such model with a specific degree of complexity for the KIParla corpus (Mauri et al., 2019). On a second level, we will show that from attempting to build such models, we learn something about the conceptual complexity of our data and workflows.

The structure of joint corpus representations calls for re-examination in an ever-changing research landscape, for several reasons: The diversity of annotation structures has increased. There are many well-established, more traditional annotation tools with specific, but distinct requirements on their tokenization (e. g. Straka et al., 2016; Schmid, 1999). Large Language Models (LLMs) are increasingly used for annotation (e. g. Huang et al., 2025; Imamovic et al., 2024) and cannot guarantee structured output, thus integration steps are required. Data collection incorporates massive amounts of non-standard and spoken corpus data, which introduces new sources of noise on many levels, such as orthography or segmentation. Multiple search platforms and corpus ecosystems exist, which makes corpus migration or alternative representation a scenario of consideration. And last but not least, increased awareness and coded criteria for research data sustainability foster reuse and extension of linguistic data sources, which demands a workflow for joint search on an increasing amount of annotation for the same data, as well as the representation of theoretically competing concepts.

One such setting is the integration of externally available corpus data for register research into an existing corpus collection and the underlying corpus ecosystem. *Register* refers to "those aspects of socially recurring intra-individual variation in linguistic behavior that are influenced by situational and functional parameters" (Lüdeling et al., 2024b). As the number of such parameters and their combinations overall is very high. Thus, to identify and reliably track and contrast features of the situation in which language is produced and the functions expressions may have, integrating new data is a directly motivated enterprise. This can mean migrating corpus data into already available data models,

[1] By this we mean the corpus data that the user indicated their interest in. For a fixed selection of corpora or corpus data, the user searches a unified corpus.

but also to integrate ever-new annotation layers to fully understand which parameters influence linguistic variation, how, and when. This complex (infra-)structural challenge is, on many levels, central to our paper. The specific use case discussed in this paper is the integration and architectural extension of the KIParla corpus (Mauri et al., 2019).

We start by highlighting important aspects and properties of the KIParla corpus and its relevance to register research in Section 2, which motivates its integration into a different ecosystem in the first place and forms the requirement for representing it as a corpus graph. We continue by looking at joint *graph* representations in Section 3 and by deriving complexity classes within the family of such models in Section 4. Section 5 connects conceptual and infrastructural aspects of corpora to such classes. In Section 6 we briefly describe a tool required to efficiently obtain graph models of any of the given classes. In Section 7 we analytically derive and present a corpus model for KIParla. We summarize our findings in Section 8.

2 The KIParla Corpus

2.1 Design Principles

KIParla is a continuously expanding corpus of spoken Italian designed to capture interactional language across a wide range of communicative settings (Mauri et al., 2019). Its architecture is guided by three core principles: ecological validity, interactional richness, and sustainability of corpus growth.

First, the corpus prioritizes naturally occurring speech, avoiding experimental elicitation whenever possible. This results in data that preserve the temporal, sequential, and collaborative properties of spoken interaction, including phenomena such as overlaps, interruptions, hesitations, and co-construction of utterances. Second, KIParla is explicitly designed to represent interaction as a primary object of analysis. Rather than reducing speech to linearized textual sequences, it encodes the multi-layered structure of dialogue, where multiple speakers produce partially overlapping and temporally interdependent contributions. This makes the corpus particularly suited for research on interactional linguistics, pragmatics, and register variation.

Third, and crucially, KIParla is designed from the outset as an incrementally growing, modular resource. New data are systematically integrated as modules, each defined by a coherent interactional setting, sampling strategy, and balancing criteria. Modules have been developed and released at different moments, reflecting the progressive expansion of the corpus and its research agenda. Current modules cover in total 3.6 million token and 350 hours of recordings to be investigated for register variation in academic contexts (lectures, exams, interviews, free conversations), semi-structured sociolinguistic interviews, Italian spoken by speakers with an immigration background in multilingual interactional contexts, as well as informal conversations in domestic settings.

Each module contributes a dataset characterized by relatively homogeneous situational parameters (e. g. degree of formality, participant roles, interactional goals, sociolinguistic profiles), while the corpus as a whole captures structured variation across contexts. This modular organization enables both within-module comparability and cross-module analyses of register and sociolinguistic variation.

Importantly, this architecture has direct consequences for corpus modeling. Because modules differ systematically in interactional properties—such as overlap density, turn-taking organization, and temporal alignment—the incremental addition of modules increases the structural heterogeneity of the corpus. In this sense, KIParla's modular design is not only a strategy for corpus growth, but also a key factor in the emergence of the representational challenges addressed in this paper. Furthermore, it makes the corpus particularly attractive for incremental register investigation along grammatical features and phenomena, integrating new perspectives as the corpus grows.

2.2 Data Format and Annotations

The KIParla corpus is available in multiple annotation formats: ELAN[2] (Wittenburg et al., 2006) files for time-aligned, multi-tier representations of spoken interaction; a csv-format for extensive metadata on both the speakers and recorded conversations (e. g. age, gender, education, linguistic background, interaction type, recording context). For corpus querying and distribution, ELAN annotations are systematically transformed into a custom tabular format (tsv, illustrated in Appendix A), in which each row corresponds to a token and each column encodes a specific annotation layer. This tabular

[2]Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands, https://archive.mpi.nl/tla/elan.

representation is optimized for compactness, extensibility, and efficient querying (Pannitto and Mauri, 2026). Importantly, this transformation is lossless: all information encoded in the ELAN files is preserved in the tabular format. However, the nature of the representation changes from an explicitly structured to a structurally implicit encoding. While ELAN encodes relational and temporal structures explicitly through tier organization and time alignment, the tabular format encodes part of this structure indirectly, through string-based annotations and index references. This format will be the only one considered further on.

Each token in the tabular format is associated with a rich set of annotations derived from Jefferson-style transcription conventions (Jefferson, 2004), capturing fine-grained properties of spoken interaction. These include speaker identification, enabling the reconstruction of dialogic structure; transcription units (TUs), representing interactionally meaningful segments; two transcription layers, consisting of the original Jefferson-style transcription with inline interactional and prosodic markup (span), and a normalized version without such markup (form); token classification, distinguishing linguistic material from non-linguistic events (e. g. pauses); prosodic and temporal features, such as prolongations and speech rate variation; alignment information, linking tokens to time spans in the audio signal; and overlap annotations, encoding simultaneous speech events across speakers.

A crucial property of this representation is that many annotations are encoded as references to character spans within tokens. For example, overlap and prosodic features are not represented as independent structural objects, but as indices referring to substrings of the token form. This design yields a compact and flexible encoding, while requiring that structural relations be reconstructed from the encoded information for downstream modeling.

From a modeling perspective, the data thus consist of partially ordered speech events, where linear order is well-defined within speakers but only partially specified across speakers due to overlaps. These properties are fully encoded in the data, but not always directly represented as explicit relational structures in the tabular format.

Finally, the format is designed with future extensibility in mind. Additional annotation layers—such as syntactic dependencies (e. g. Pannitto et al., 2025)—can be integrated without altering the core structure of the data.

2.3 Central Goals and Desiderata

The extension of KIParla to additional corpus ecosystems is motivated by both practical and theoretical considerations. From a practical perspective, the goal is to increase accessibility and interoperability. While KIParla is natively available in its tabular format optimized for specific tools, making it available in a graph-based environment allows a broader community to access and explore the data using a different query paradigm. From a theoretical and modeling perspective, the transformation process aims to achieve three main objectives:

Maximum preservation of information All annotations encoded in the original data should be retained without loss. This includes both explicit annotations and implicit structural information encoded via string references.

Structural explicitness Wherever possible, information that is encoded as strings or indices in the source format should be reinterpreted as explicit structures in the target representation. For example, overlap relations should not remain encoded as character offsets, but should be represented as relational structures between linguistic units.

Robustness and extensibility of the processing pipeline The transformation workflow should be able to handle future extensions of the corpus—both in terms of new data and new annotation layers—without requiring substantial reconfiguration. Ideally, the pipeline should support automated validation and reproducible corpus builds.

These desiderata reflect a broader commitment to sustainable corpus development, where data are not only preserved but also made adaptable to evolving research questions and technological environments. The observations from Section 1 about a diversified research environment make such desiderata necessary and, at the same time, render their achievement a complex endeavor.

3 Joint Corpus Graph Representations

Graphs are powerful mathematical structures capable of representing discrete linguistic data, thus a graph-based model allows for the representation of basically any annotation type or structure. As elaborate formal representations, they also allow us to reason about features and types of linguistic corpora, i. e., their complexity, composition, and subclasses. For the purpose of this paper, for many

practical reasons we choose *graphANNIS* (Krause, 2019), but argue, that all points laid out here generalize beyond this specific representation. They are features of corpora and can be considered requirements for adequately representing data with specific properties. As such, *corpora* are considered collections (i. e., finite sets) of discrete items or events, whose linguistic features are represented via nodes and edges, and labels attached to both. In the discreteness of linguistic items lies the central limitation of this approach to generality across all linguistic data, but for the majority of corpora and their analyses this is negligible.

graphANNIS is not only a database backend for corpus search interfaces, with its central application being the backend of ANNIS (Krause, 2019). It is also a graph data model and thus a modeling framework for linguistic corpora, that provides strong means to represent and search multi-layer corpora, limited mostly in physically available space and restriction to acyclic structures. As any graph, a corpus graph consists of nodes and edges, where the nodes are either corpus-structural nodes (type CORPUS) or linguistic, not necessarily overt nodes (type NODE). Both node types can hold annotations (or more formally, *properties*). While annotations on the former are usually considered metadata, annotations on the latter are linguistic annotations.

In graphANNIS and our model, edges are always directed and can also hold annotations. Edges are distinguished into types as well and organized in *components*, which are sets of edges that share the component type and a component identifier, to allow using an edge type for multiple components without introducing cycles to the graph. Each edge type corresponds to one or more search operators in the *ANNIS Query Language* (AQL), which can express levels 1, 2, and 3 of the CQLF meta model (Evert et al., 2025). As this implies highly expressive representations, we take this as an additional argument that our characterizations of corpora throughout this paper generalize beyond the given type of graph model.

In which way a component is read off when a corpus graph is queried, is described as *search semantics* in the following. For the modeling discussion of this paper we only need to look at three edge types:

ORDERING Edges of this type are used to linearly order elements, usually segments of a data source, in the annotation graph. They usually form a non-branching tree, i. e., there is a single path from one root to one terminal node for each connected sub component. A corpus graph can contain more than one ordering. Search semantics are transitive along COVERAGE edges (s. below), but only optionally transitive within the edge type.[3] As an example, the linear ordering of the tokens of a corpus is realized by such a component, with one token corresponding to a node, an outgoing edge from said node pointing to its successive token, and an incoming edge pointing from the preceding token. Additional orderings can be used to order alternative text segmentations, which will be discussed later. An example is given in Figure 1.

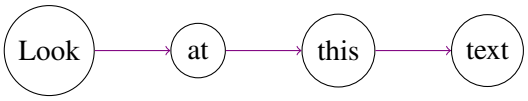

Figure 1: ORDERING edges providing a linear order to token nodes in a graphANNIS corpus model.

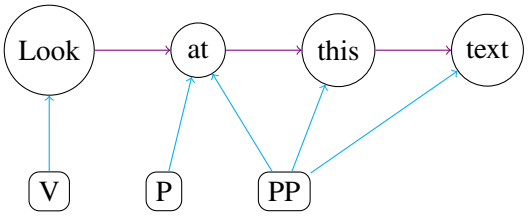

Figure 2: ORDERING edges (violet) and COVERAGE edges (blue) in a corpus graph.

COVERAGE These edges are used to express association of linguistic annotations or features with parts of the text, directly or indirectly. Edges of this type have *undirected* and transitive search semantics. In Figure 2 the annotation "P" overlaps with "at", "PP" with "at this text", and hence annotation "P" partially with annotation "PP" due to transitivity and lack of direction. These semantics make search efficient, but also restrict the use of COVERAGE edges as pointed out below in Sections 4.2 and 4.3. Elaborating a bit more on transitivity of ORDERING edges along COVERAGE edges as mentioned above, we point out that in Figure 2 the node with annotation "P" is a successor of the node with annotation "V", as their covered tokens are adjacent in the ordering.

POINTING Semantically, this is the most underspecified edge type. The most common applications are dependency syntax (conceptually hierarchical) and referential chains (conceptually non-

[3] I. e., there is a transitive and an intransitive AQL operator.

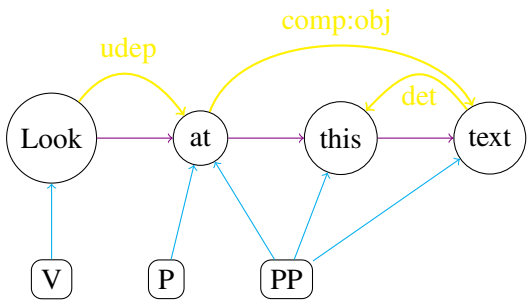

Figure 3: ORDERING edges (violet), COVERAGE edges (blue) and POINTING edges (yellow) in a corpus graph. The syntactic dependencies follow Gerdes et al. (SUD, 2019).

hierarchical). Search semantics are directed and optionally transitive. Figure 3 extends the previously given examples by POINTING edges.

Any two nodes can be connected via an edge in any possible component without further restrictions. It is the user's responsibility to only build sensible graphs or avoid cycles. Further structural and semantic restrictions to graphANNIS models can be imposed by the search engine.

4 Complexity Classes

With only the described node and edge types, we will describe three classes of corpus graph models, whose structural features reflect an underlying conceptual complexity of the represented corpora. Hence, we consider them complexity classes of corpus graph models, where a *graph model* is a graph representation of a **specific** corpus. The distinction into the provided classes is derived from requirements of soundness and stability (Shadrova et al., 2025, 88–93), which imposes structural restrictions on models by their semantics. As COVERAGE edges are transitive and associate annotations semantically with linguistic items, joint representations of such objects need to make sure that the association is valid: No new invalid associations are introduced in a unification (soundness) and no existing valid associations are removed (stability). The classes discussed below are a consequence of opting for stability and the expressive power of COVERAGE edges. For a specific corpus, soundness might only be given in the highest class of complexity.

4.1 SINGLESEG: Single Tokenization

The simplest corpus representation is a sequence of ordered nodes in a single default ordering component. These ordered nodes are also terminal nodes in all COVERAGE components of the model. The examples in Figures 1, 2, and 3 are single segmentation models. The ordered nodes, the nodes covering them, and the involved edges carry all annotations. The model can be enriched with other edge components and annotations. As long as no alternative ordering or segmentation is provided such that none of the orderings is minimal, a corpus has a candidate model in class SINGLESEG. As an example, an ordering of character segments paired with an ordering of the words formed from those characters is still possible within a SINGLESEG model, as the character tokens are the minimal ordering (no character is associated with more than one word). In contrast, the original spelling of historical text paired with a normalized spelling cannot guarantee minimality of one of the two segmentations, which both have their own ordering, as in one case a historically joint word might be split in the normalized segmentation, and vice versa (s. also Odebrecht et al., 2016).

4.2 MULTISEG: Multiple Segmentations

Corpus models in class MULTISEG, i. e., corpora with multiple segmentations allow for competing tokenizations or the assignment of annotations to independent text segments, i. e., the existence of overlapping, independently ordered, but indirectly aligned annotatable linguistic items. Conceptually, such corpora are supported by ANNIS, but also editors such as EXMARaLDA (Schmidt and Wörner, 2014) or ELAN and their built-in search functionality. Odebrecht et al. (2016) provide a motivation of a multiple segmentation model for the RIDGES corpus. Multiple segmentations enable for the representation of dialogue corpora with independent search by speaker, such as BeMaTaC (Sauer and Lüdeling, 2016), but also corpora adding re-segmenting normalization layers, such as RUEG (Klotz et al., 2024; Lüdeling et al., 2024a).

Each of the multiple segmentations is a set of items to its own set of annotations. In such a model, all segments point towards token nodes, that are demoted to timeline objects. Among the distinct ordered segmentations, none is minimal with respect to the others. In fact, models, whose segmentations share the same minimal nodes, can be represented as a model of class SINGLESEG. For multiple segmentations, the timeline sequence nodes provide the minimal anchor nodes to represent indirect alignments between the segmentations via COVERAGE edges. An illustration of a simplified example is

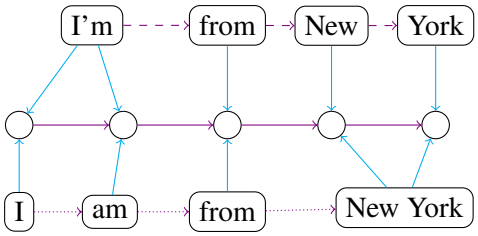

Figure 4: A MULTISEG graph model. COVERAGE edges are blue, ORDERING edges are violet. Line patterns distinguish component identifiers. Two segmentations are aligned via ordered, aligning nodes.

given in Figure 4: Both ordered sequences are locally minimal ("I'm" vs. "I" and "am", "New" and "York" vs. "New York"), but none is globally.

4.3 ALIGNSEG: Parallel Segmentations

While modeling data as multiple segmentations is more powerful, it also has certain requirements that need to be met for being realizable. Indirect alignments must be defined or derivable for all pairs of segmentations to keep the data sound. To give a simple and general example, multilingual corpora that require a reordering of tokens and provide between-language alignments (traditionally referred to as *parallel corpora*) cannot be represented as multiple segmentations, as their alignment cannot be represented indirectly through a common timeline tokenization. They require a strictly parallel model, i. e., direct, **explicit** alignments, which is, as we will show below, the most general and powerful corpus architecture and yields the most expressive models. It is often also required for observationally non-parallel data. Figure 5 provides an example graph: There is no information about the mapping between the syllables of "everyone's" ("ɛv", "ɹi", and "wʌnz") and the normalized representation "everyone" and "is". We need explicit alignments.

Going backwards from the most complex corpus models discussed, those in class ALIGNSEG, we see that we get to MULTISEG by removing the requirement for non-associative POINTING edges. At the same time, a corpus model in ALIGNSEG is not excluded from featuring multiple segmentation subgraphs, that are connected via such edges. And in the same spirit, a corpus model in SINGLESEG is the most simple corpus model type of any of the three classes. Therefore,

$$\text{SINGLESEG} \subset \text{MULTISEG} \subset \text{ALIGNSEG}$$

holds. The model class thus indicates the lower bound of complexity.

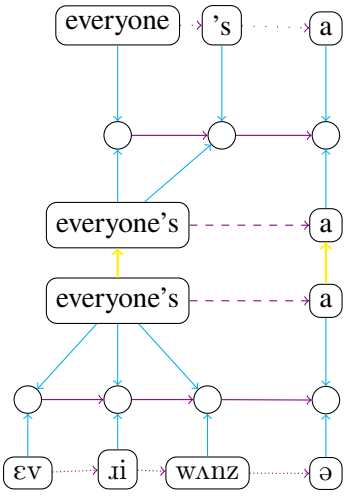

Figure 5: An ALIGNSEG graph model (Shadrova et al., 2025; Zerbian et al., 2023). COVERAGE edges are blue, ORDERING edges are violet, POINTING edges are yellow. Line patterns distinguish component identifiers. The syllabic segmentation on the bottom and the normalized segmentation on the very top suffer from under-specified simultaneity, therefore alignment edges are necessary.

5 Under-Specified Simultaneity

For determining whether a specific corpus model can be of class MULTISEG or rather ALIGNSEG, the degree of specification in simultaneity of item sequences is decisive. We discuss different cases of simultaneous item sequences, that do and do not shift a corpus from MULTISEG to ALIGNSEG to better illustrate their relevance and underlying reasons of complexity. Neither the number of data formats, that a corpus representation is derived from, nor the modality involved, nor the question whether or not a corpus contains dialogue data, necessarily determines the complexity required for representing all involved ordered sequences.

Many, if not most, linguistically-annotated corpora represent their annotations in more than one data format, which implies some sort of distributed source representation. Requirements for this are partially discussed in Shadrova et al. (2025, 89-90). The bottom line is that not all formats or the tools behind them are capable of representing just any type of annotation, but are specialized, usually as a consequence of the annotations' structures. A model of class ALIGNSEG is motivated, when soundness cannot be ensured differently. i. e., no format contains the globally minimal segmentation while the annotations target single segments, so the segmentation is actually necessary. As we will show below in our discussion on KIParla, even a

single format can raise that issue. At the same time, multiple formats can—and often do—contain the same segmentation. Thus, the number of data formats involved is neither sufficient nor necessary to predict the complexity of a corpus model.

Regarding involved modalities in corpora, spoken data are certainly more prone to produce competing simultaneous item sequences. Raw tokens from speech are usually more noisy, while processing output of tools for such data often involves some normalization. Also, speech events do not necessarily need to be bound to words. However, written data on its own can yield problems relating to under-specified simultaneity as well, even without being a parallel corpus in the original sense, i. e., without aligning data from two or more distinct languages. Extensions of the RUEG corpus contain multiple segmentations—raw and orthographically normalized tokens—linked to word formation-trees over morphemes as smallest units, whose mapping to one of the multiple segmentations is under-specified. At the same time, not all spoken corpora require a model more complex than multiple segmentation.

Figure 5 is an example for spoken data from Zerbian et al. (2023), a prosodically annotated subcorpus of the RUEG corpus (Zerbian et al., 2023). It contains morphosyntactic annotations on normalized tokens with mappings to raw forms, and prosodic annotations on syllables mapped to raw forms. The data set is missing a crucial mapping from normalized forms to syllables, which cannot be derived transitively (Shadrova et al., 2025, 90-93). In other words, while the the morphological and the prosodic model are both in class MULTISEG, their unification or joint representation cannot be, as no indirect alignment can be obtained between two simultaneous item sequences.

Dialogue data can also yield MULTISEG models. BeMaTaC (Sauer and Lüdeling, 2016) presents a corpus, that has simultaneous segmentations of an instructor and an instructee, but fully specified, such that a MULTISEG model can be derived.

The central question in determining the complexity required for a corpus model is whether the relationship between two ordering components can be sufficiently derived from a common minimal tokenization such as a timeline. If this is not the case, explicit between-item alignment is required. Therefore, any corpus who relies on a plurality of item sources has potentially, but not inevitably, a graph model in ALIGNSEG.

6 Building Graph Models

For building graph models, also the one for the KIParla corpus outlined below, we require the novel tool *Annatto* (Krause and Klotz, 2026). Annatto is a compilation tool to build graphANNIS models for corpora from linguistic data. It adopts Pepper's (Zipser and Romary, 2010) support for a multitude of formats and adds more configurable, lower-level graph manipulations and compile-time checking of linguistic corpora for increased genericity, efficient transformation, and quality assurance. Annatto implements format mapping to graphANNIS instead of Salt graphs. This way, a corpus graph under manipulation can be queried at build-time to identify relevant parts for manipulation and can be checked for consistency and other aspects of corpus quality long before being imported into a search engine. Both aspects guarantee more efficient corpus building processes and further allow to automatically build and test corpora in continuous integration pipelines.

Low-level graph manipulations are realized as *graph operations*. They are a trade-off between the lowest degree of graph manipulation as available in *grew* (Bonfante et al., 2018), on the one hand, and Pepper's traditional, rather coarse approach on the other. *grew* requires an explicit transformation grammar, as its central purpose is natural language processing and analysis, not corpus checking and compilation. It does not support the same amount of import formats or all corpus model classes required for this paper's purposes. Pepper's drastically less flexible manipulations, on the other end, are too rigid.

Introducing strongly configurable graph operations comes with the benefit of reduced pressure on format support, as long as the most generic formats are supported. This way, importing a specific format can be implemented by a generic import and a sequence of graph operations, as applied in this paper for KIParla's table format.

7 Modeling KIParla

A graphANNIS representation is built from a single format source, but with multiple annotation files. Additionally, audio files need to be linked and aligned without further manual processing. The corpus contains temporarily overlapping linguistic items from multiple speakers. Annotations contain explicit references to character indices of other annotation values. In the following the corpus graph

is sketched by component type. An illustration of a subgraph of the corpus model is given in Figure 6 in Appendix B.

ORDERING From the given prerequisites, requirements towards different components can be derived. A non-trivial ordering for sub-word level items, which could be characters, is required. This is due to the back-reference of annotations, such as coded overlaps, to character indices of forms, but also due to the semantics of the overlaps themselves, i. e., partial temporal overlap between items from different speakers. If one token of speaker 1 and another token of speaker 2 partially overlap, it is crucial to not cross-associate annotations of speaker 1's token(s) with the entirety of speaker 2's overlapping token(s) if this is not what happened in the input. The model needs to be sound. Additionally, the individual speaker's forms need explicit ordering, as it is given only implicitly in the input.

COVERAGE We aim for structural explicitness, therefore simply attaching all annotations in KIParla's table format directly to their linguistic item is discarded. Also, above it was outlined that items below word-level are required to represent the data, thus spans over such items require edges of type COVERAGE. More crucially, though, is the question whether the KIParla data allow for multiple segmentation. As a reminder, the requirement for such a representation is total derivability of indirect alignments between all ordered segments in the model. For KIParla, those segments are items below word-level, and the word forms of each speaker. To be in a position to derive a complete common timeline (which is what deriving indirect alignments boils down to), either complete time annotations for all segments have to be available in the source data, or the overlap information derivable from the input data is sufficient for deriving them. However, time annotations are only available for transcription unit boundaries. In our experience, this is the case for most spoken corpora, that provide audio alignments, as word-level alignments imply costly manual labor or potentially faulty automatic values. Regarding the overlap annotations in KIParla, they are also not sufficient, as overlapping sections are not strictly in the ratio of $1 : n$, therefore alignments for tokens within overlapping intervals are unknown. This, as well, is very common. For KIParla we conclude, that maximum information preservation and soundness are only given for a model in ALIGNSEG.

POINTING It follows that a parallel corpus graph structure is adopted to represent the overlaps between multiple speakers. In Section 5 we argued, that this is the most generic corpus model class, that fits all corpora as a consequence of relying on under-specification. Beyond alignment, each speaker's syntactic dependencies can be integrated as labeled pointers, once they are available in the main corpus (Pannitto et al., 2025).

8 Summary and Discussion

We discussed many reasons, why a more classic MULTISEG model does not suffice for all corpora with competing orderings: As corpora get reused more frequently, and tools allow to revise an existing item source to research-specific needs while potentially wanting to incorporate existing annotations, any new investigation of a corpus is always at "risk" to be the moment of architectural revision to a more complex model. Of course, as already pointed out, the increasing role of corpora of spoken language makes such a scenario more likely, as these data are inherently more prone to competition between item sequences. Tools for efficient, but not necessarily reliable, automatic transcription, are increasingly available and integrated into corpus workflows. Such tools often provide a different data segmentation than, as an example, a dependency parser analyzing the outcome of such transcription.[4] As soon as both segmentations are required to be present in a searchable corpus, parallel modeling may be inevitable. With an ever-increasing number of linguistic tools available in general, it becomes more likely that incompatible tokenizations need unification in a corpus pipeline. The often unstructured output of LLMs will add to that as well. But not just the multitude of available tools, also the openness of corpus architectures to ever more annotations may be contributing to the described tendency.

The outlined abstract corpus representations deserves a comment in the direction of what constitutes a parallel corpus. This is not to challenge, that a parallel corpus in its traditional sense is and remains a multilingual corpus with explicit alignments. We merely argue and have shown, that models used for parallel corpora may be required for other data sources as well, and highlight that there are cases of *within-data* parallelism. Or, to look at

[4] A common example are the tokenizations of the powerful tools Whisper (Radford et al., 2023) and UDPipe (Straka et al., 2016).

it from a modeling point of view, one may have to distinguish between *syntax* (i. e., the structure) and *semantics* (the content) of a corpus when classifying it.

Enriching all of KIParla's annotations in a graph model revealed that it conceptually and structurally is of the highest class of complexity. It is a particularly special case, as it manages to code its underlying data in a single, in itself very simple format. The complexity is unpacked, once all available information is to be translated into searchable structures, which gives it an aspect of powerful and lossless compression from a corpus coding perspective. Also, on a development level, KIParla aims at a high degree of sustainability and reusibility, which falls in line with our observation that these aspects are likely to create a demand for complex corpus models. We nevertheless succeeded to compile a graph-based corpus model through a sequence of graph transformations and to build a robust compilation process including measures of quality assurance.[5] From this, a valuable corpus resource is available for register investigations in an additional corpus search environment.

Regarding future corpus projects starting from scratch or building on an existing resource, the growing tool landscape and individual research may yield similarly complex corpus models, unless the number of information sources is reduced, the information is represented in a compressed form, or segmentation procedures undergo some standardization. Compressed information representation, though, yields less ergonomic search procedures through structural implicitness. And standardized segmentation procedures usually do not serve a wide range of research interests in the context of linguistics. Beyond that, projects aiming at corpus expansion need to be aware, that in the multilayer paradigm such an endeavor is not necessarily straight forward, when new item sources are introduced.

Our discussion strictly aims to generalize beyond the context of graphANNIS corpora, as the model itself is very generic, as are graph representations. Conceptual complexities revealed by such analyses thus need to be addressed in different frameworks and search engine contexts as well, when the goal is a rich representation of all available information. The derived complexity hierarchy of corpus models in Secion 4 can be a tool to conceptually reason about corpora also outside of the ANNIS ecosystem. Even when **not** working with unified corpora or **not** querying joint representations, the issues outlined in the modeling section have high relevance. Simply thinking of a corpus' graph representation or formally modeling it in an apt framework helps identify conceptual or practical pitfalls. As soon as multiple item sources are involved, the underlying questions, that need to be answered are whether the item sources need to be evaluated *jointly* to some extent and which degree of effort is required to do so. As an example, assume a corpus is represented in two formats, one holding constituent syntax on one item source or tokenization. The other format might hold prosodic features over syllables as items. A joint graph representation of such a corpus provides at least a hint, how easily different prosodic features can be investigated across different syntactic phrase types. To avoid the wrong conclusions from this example, we remind that this work has explicitly demonstrated, among other things, that many aspects of searchable corpus representations are independent of the format or number of formats they are sourced from.

As less complex models (multiple segmentation models) can be derived from complete audio alignments in specific cases, enriching existing corpora with partial time value annotations can sometimes circumvent a parallel corpus structure. Given a sufficient degree of quality in the audio signal, tools such as WebMAUS (Kisler et al., 2017) are providing word-level alignments, as long as resources are available to include them in a corpus pipeline. Such an approach is a concrete instance of re-allocating complexity between model and workflow, which is common in corpus linguistics.

Limitations

Our discussion cannot be generalized to models with infinitely small target items (i. e., points in a continuously annotatable space). Also, alternative ways of dealing with competing orderings, such as allowing crossing COVERAGE edges and thus a free re-ordering of alternative segmentations in MULTISEG models is out of scope for this paper. Furthermore, corpora demanding explicit alignments can be represented as MULTISEG models when ordering semantics are weakened. This is possible by having sequences of exclusive alignment nodes for each segmentation. Under strict ordering se-

[5]The workflow for a prototype model is available here: https://doi.org/10.5281/zenodo.21640716

mantics as discussed in this paper, approaches of exclusive alignment or flexible re-ordering are not possible. Thus, the discussed complexity hierarchy holds. In general, our discussion was outlined along specific edge properties and semantics. If we were to modify them, the complexity would not simply disappear, but get re-allocated within the model or into the search algorithm. Such modifications are beyond what can be discussed here.

Acknowledgments

We express our gratitude to three anonymous reviewers for their valuable comments and feedback.

This research was funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) – SFB 1412, 416591334 and FOR 2537, 313607803, GZ LU 856/16-1.

References

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Nils Diewald, Michael Hanl, Eliza Margaretha, Joachim Bingel, Marc Kupietz, Piotr Bański, and Andreas Witt. 2016. KorAP architecture — diving in the deep sea of corpus data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3586–3591, Portorož, Slovenia. European Language Resources Association (ELRA).

Stephanie Evert, Timm Weber, Steffen Bothe, Philipp Heinrich, and Alexander Piperski. 2025. *Data exploitation: corpus queries*, pages 303–338. De Gruyter.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving Surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 126–132, Paris, France. Association for Computational Linguistics.

Ding Huang, Jiajin Xu, Yingming Song, and Ruchen Yu. 2025. Assessing the potential of using large language models for pragmatic annotation of historical texts. *Journal of Historical Pragmatics*, 26(3):406–437.

Mirela Imamovic, Silvana Deilen, Dylan Glynn, and Ekaterina Lapshinova-Koltunski. 2024. Using ChatGPT for annotation of attitude within the appraisal theory: Lessons learned. In *Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 112–123, St. Julians, Malta. Association for Computational Linguistics.

Gail Jefferson. 2004. Glossary of transcript symbols with an introduction. In Gene H. Lerner, editor, *Pragmatics & Beyond New Series*, volume 125, pages 13–31. John Benjamins Publishing Company, Amsterdam.

Thomas Kisler, Uwe Reichel, and Florian Schiel. 2017. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326–347.

Martin Klotz, Rahel Gajaneh Hartz, Annika Labrenz, Anke Lüdeling, and Anna Shadrova. 2024. Die RUEG-Korpora: Ein Blick auf Design, Aufbau, Infrastruktur und Nachnutzung multilingualer Forschungsdaten. *Zeitschrift für germanistische Linguistik*, 52(3):578–592.

Thomas Krause. 2019. *ANNIS: A graph-based query system for deeply annotated text corpora*. PhD Thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät.

Thomas Krause and Martin Klotz. 2026. Annatto.

Kunilovskaya, Maria and Koviazina, Marina. 2017. Sketch Engine: A Toolbox for Linguistic Discovery. *Journal of Linguistics/Jazykovedný casopis*, 68(3):503–507.

Anke Lüdeling, Artemis Alexiadou, Shanley Allen, Oliver Bunk, Natalia Gagarina, Sofia Grigoriadou, Rahel Gajaneh Hartz, Kateryna Iefremenko, Esther Jahns, Kalliopi Katsika, Mareike Keller, Martin Klotz, Thomas Krause, Annika Labrenz, Maria Martynova, Onur Özsoy, Tatiana Pashkova, Maria Pohle, Judith Purkarthofer, and 10 others. 2024a. RUEG Corpus.

Anke Lüdeling, Luka Szucsich, Lars Erik Zeige, Aria Adli, Artemis Alexiadou, Malte Belz, Miriam Bouzouita, Oliver Bunk, Malte Dreyer, Markus Egg, Anna Helene Feulner, Jürg Fleischer, Natalia Gagarina, Aron Hirsch, Stefanie Jannedy, Pia Knoeferle, Thomas Krause, Silvia Kutscher, Mingya Liu, and 18 others. 2024b. Register: Language Users' Knowledge of Situational-Functional Variation. Proposal frame text, CRC 1412 Register, Phase II. *REALIS: Register Aspects of Language in Situation*, 3.

Caterina Mauri, Silvia Ballarè, Eugenio Goria, Massimo Cerruti, and Francesco Suriano. 2019. KIParla corpus: A new resource for spoken Italian. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, pages 243–249, Bari, Italy. CEUR Workshop Proceedings.

Carolin Odebrecht, Malte Belz, Amir Zeldes, Anke Lüdeling, and Thomas Krause. 2016. RIDGES Herbology - Designing a Diachronic Multi-Layer Corpus. *Language Resources and Evaluation*.

Ludovica Pannitto and Caterina Mauri. 2026. Reuse by Design: A Pivot-Based Architecture for the KIParla Corpus of Spoken Italian. *Journal of Open Humanities Data*.

Ludovica Pannitto, Eleonora Zucchini, Silvia Ballarè, Cristina Bosco, Caterina Mauri, and Manuela Sanguinetti. 2025. Introducing KIParla forest: seeds for a UD annotation of interactional syntax. In *Proceedings of the Eighth International Conference on Dependency Linguistics (Depling, SyntaxFest 2025)*, pages 54–73, Ljubljana, Slovenia. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Simon Sauer and Anke Lüdeling. 2016. Flexible multilayer spoken dialogue corpora. *International Journal of Corpus Linguistics*, 21(3):419–438.

Helmut Schmid. 1999. Improvements in Part-of-Speech Tagging with an Application to German. In Susan Armstrong, Kenneth Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, pages 13–25. Springer Netherlands, Dordrecht.

Thomas Schmidt and Kai Wörner. 2014. EXMARaLDA. In Jacques Durand, Ulrike Gut, and Gjert Kristoffersen, editors, *The Oxford Handbook of Corpus Phonology*. Oxford University Press.

Anna Shadrova, Martin Klotz, Rahel Gajaneh Hartz, and Anke Lüdeling. 2025. Mapping the mappings and then containing them all: Quality assurance, interface modeling, and epistemology in complex corpus projects. In Shanley Allen, Mareike Keller, Artemis Alexiadou, and Heike Wiese, editors, *Linguistic Dynamics in Heritage Speakers*.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN : a professional framework for multimodality research. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1556–1559, Genoa. 5th International Conference on Language Resources and Evaluation (LREC 2006).

Sabine Zerbian, Yulia Zuban, and Martin Klotz. 2023. Intonational Features of Spontaneous Narrations in Monolingual and Heritage Russian in the U.S.—An Exploration of the RUEG Corpus. *Languages*, 9(1):2.

Florian Zipser and Laurent Romary. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Workshop on Language Resource and Language Technology Standards, LREC 2010*, La Valette, Malta.

A String-Based Representation of Annotations in KIParla

As discussed in Section 2, the tab-separated format of KIParla represents structural information as string codings. An example snippet is given in Table 1. The table only represents a subset of the actually available annotations. The example contains speakers `BO026` and `BO020`. The speaker is coded in column *speaker*, the orthographically normalized token is given in column *form*, the identifier of the transcription unit (a syntactic grouping) in column *TU* and a string coding of overlaps in column *overlap*. Tokens `36-0` to `36-3` represent the tokens of a transcription unit of the first speaker, followed by parts of a transcription unit of the other speaker.

While the format enforces a sequential order, the overlap column indicates that the tokens `36-1`, `36-2`, `37-0`, and `37-1` partially overlap in the audio signal. To understand this, the overlap codings need to be unpacked. They code the character index range—understood as corresponding to underlying phonetic segments—starting at 0 with an exclusive right boundary, which is combined with an overlap identifier in parentheses. Thus, for speaker `BO026` character 6 of token *partite* (the final *e*) and all characters of token *da* (`36-2`) participate in overlap `10`. For speaker `BO020`, all characters of tokens *sì* and *da* participate in the same overlap, i. e., the sequences *e da oggi* and *sì da* are uttered simultaneously. There is no further statement in the data on the correspondence between the smallest units, the characters, to one another *between* two speakers for a specific overlap.

token_id	speaker	form	TU	overlap
		…		
36-0	BO026	sarebbero	36	
36-1	BO026	partite	36	6-7(10)
36-2	BO026	da	36	0-2(10)
36-3	BO026	oggi	36	
37-0	BO020	sì	37	0-2(10)
37-1	BO020	da	37	0-2(10)
		…		

Table 1: The string representation (tsv) of KIParla at the example of a snippet of situation `BOA1001`. "…" mark omissions. Not all annotation columns are represented for space reasons.

B The Graph Representation of KIParla

Figure 6 shows a graph model derived from the data in Table 1. The upper half of the graph is the MULTISEG graph of speaker (`B0026`), the lower half of speaker `B0020`. The overlap nodes cover the overlapped parts of the production via COVERAGE edges. The overlap annotations in each subgraph are linked via alignment POINTING relations. "…" indicate omitted subsequences of ORDERING components. `transcription unit` annotations—which are structurally implicit token-by-token annotations in Table 1—are **explicit** single structures in the graph model. Annotation values are omitted. Structural explicitness is expanded to other annotations as well, but they are omitted in the example. Note that annotations are not explicitly ordered, as they are not segments.

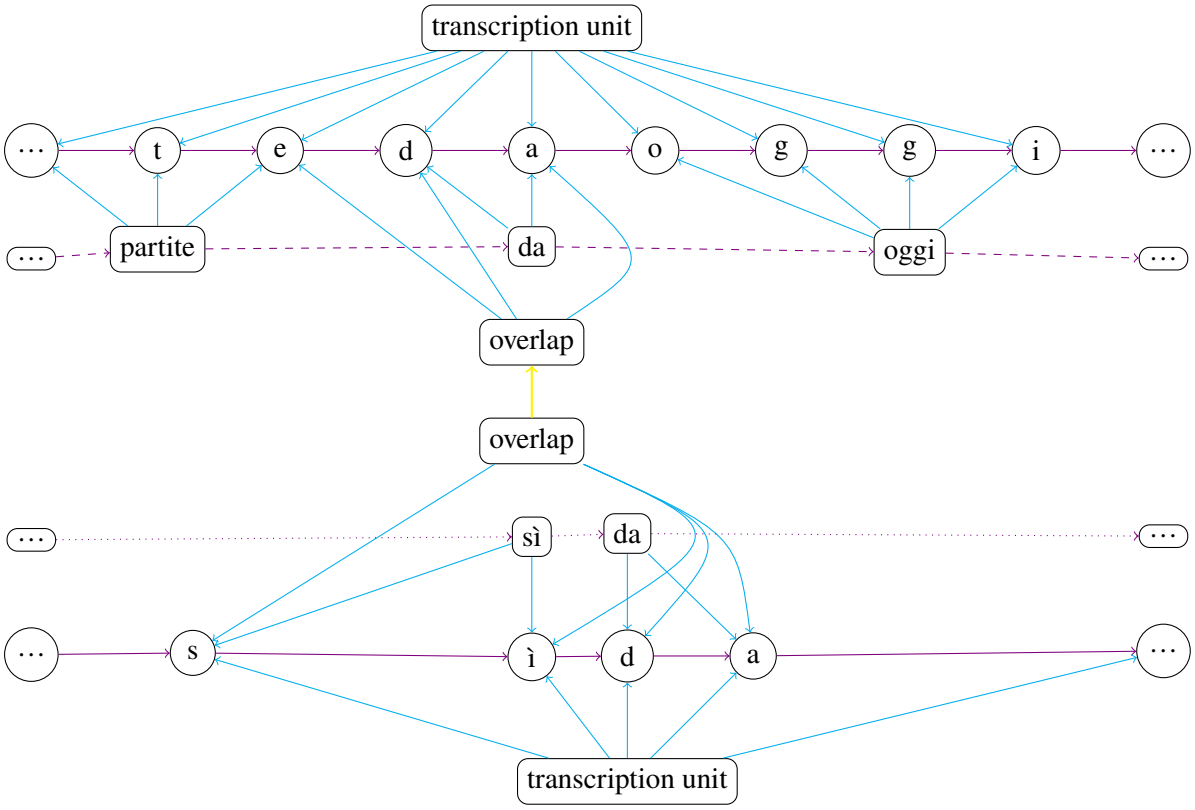

Figure 6: An ALIGNSEG graph model for KIParla at the example of situation `B0A1001` from the KIP module. COVERAGE edges are blue, ORDERING edges are violet, POINTING edges are yellow. Line patterns distinguish component identifiers.