Designing annotation schemes: From model to representation

Nancy Ide, Christian Chiarcos, Manfred Stede, and Steve Cassidy

Abstract The physical formats used to represent linguistic data and its annotations have evolved over the past four decades, accommodating different needs and perspectives as well as incorporating advances in data representation generally. This chapter provides an overview of representation formats with the aim of surveying the relevant issues for representing different data types together with current state-of-the-art solutions, in order to provide sufficient information to guide others in the choice of a representation format or formats.

1 Introduction

Historically, designers of linguistic annotation schemes have focused on determining the appropriate categories and features to describe the phenomenon in question (as described in Chapter 2) and paid less attention to the eventual *physical representation*, or *representation format*, of the annotation information. In fact, the separation between conceptual content and physical representation has not always been taken into account when schemes are designed, with possibly unintended results when constraints imposed by the physical representation affect choices for the conceptual content of an annotation scheme; for example, a representation format may impose limits on the complexity of the information that can be included or force the conflation of information into cryptic labels, which may later prove to be undesirable. In recent years, the need to compare and combine annotations as well as use them in software environments for which they may have not been originally designed has increased, leading to the awareness that a conceptual scheme may be represented in any of a variety of different physical formats and/or transduced from one to the other, and therefore, that interactions between the design of a conceptual scheme and physical format not only can, but also should be avoided.

Steve Cassidy
Department of Computing, Macquarie University, Sydney e-mail: Steve.Cassidy@mq.edu.au

This chapter provides an overview of representation formats with the aim of surveying the relevant issues for representing different data types together with current state-of-the-art solutions, in order to provide sufficient information to guide others in the choice of a representation format or formats. We begin with a historical account of their evolution over the past 25-30 years (Section 2) and cover the representation issues for text (Section 3) and multi-modal data (Section 4). We then provide examples of state-of the art representation schemes (Section 5) intended to generalize over a wide range of annotation types, including graph-based schemes and representation of linguistically-annotated resources as linked data, and additional concerns and possibilities such as querying and linking to ontologies and other resources. The chapter concludes by providing practical guidance for choosing a representation for linguistically annotated data (Section 6).

2 Background

A physical representation performs one or more of several functions, depending on the type of annotation. First and foremost, a representation format must provide means to associate linguistic information with regions of the data being annotated. This information typically consists of annotation *labels* (i.e., identifiers indicating what the data in the region is, in linguistic terms–e.g., token, utterance, noun chunk, verb phrase, morpheme, disfluency, person, etc.) and may also specify linguistic or other relevant *features* of the data (e.g., root/lemma, duration or prosodic characteristics for speech data, sense tag, etc.). Where necessary, the representation may also enable specification of *relations* between annotated items, including structural relations (e.g. parent-child in a constituency parse tree), functional relations (coreference, temporal, dependency, etc.), and in some cases, simple component connections (e.g., discontiguous parts of a linguistic entity).

The primary concern in determining format, especially in the 1980s and early 1990s, was the ease of processing by software that would use the output. For example, early formats for phenomena such as part of speech often output one word per line, separated from its part of speech (POS) tag by a special character such as an underscore or slash [1, 2]. Syntactic parsers producing constituency analyses typically used what has come to be known as the "Penn Treebank format", which brackets and nests constituents with parentheses, LISP-style [3, 4, 5] (see section 3.2.1). Dependency parsers often used a line-based format that provides the syntactic function and its arguments in specified fields (see Chapter IV, section 5, for a detailed description). Interestingly, these early formats for POS tagger and parser output have remained in use, with very little variation, up to the present day, primarily in the output of POS taggers; see for example, the Stanford taggers and parsers

for multiple languages[1], TreeTagger[2], and TnT[3]. Such formats rely heavily on white space and line breaks, together with occasional special characters, to delineate elements of the analysis (e.g., individual tokens and part of speech tags). As a result, software intended to use these formats as input must be programmed to understand the meaning of these separators, together with the nature of the information in each field.

Over the past 30 years, generalized solutions for representing annotated language data–i.e., solutions that can apply to a wide range of annotation types and therefore allow for combining multiple layers and types of linguistic information–have been proposed.[4] The earliest format of note is the Standard Generalized Markup Language (SGML; ISO 8879:1986) [6], which was introduced in 1986 to enable sharing of machine-readable documents, with no special emphasis on (or even concern for) linguistically-annotated data. Like its successor, the Extensible Markup Language (XML) [11], SGML defined a "meta-format" for marking up, or annotating, electronic documents consisting of rules for separating markup (tags) from data (by enclosing identifying names in angle brackets) and providing additional information in the form of attributes (features) on those tags.[5] SGML also specified a context-free language for defining tags and the valid structural relations among them (nesting, order, repetition, etc.) in an *SGML Document Type Definition* (DTD) that is used by SGML-aware software to validate the appropriate use of tags in a conforming document. XML replaced the DTD with the XML schema, which performs the same function as well as some others.

The Text Encoding Initiative (TEI)[6] Guidelines, first published in 1992, defined a broad range of SGML (and later, XML) tags and accompanying DTDs for encoding language data. However, the TEI was from its beginnings intended primarily for humanities data and does not provide guidelines for representing many phenomena of interest for linguistic annotation. Therefore, in the mid-1990s, the EU EAGLES project[7] defined the Corpus Encoding Standard (CES) [23], a customized application of the TEI providing a suite of SGML DTDs for encoding linguistic data and annotations, which was later instantiated in XML (XCES) [7]. In part as a result, SGML (and later, XML) began appearing in annotated language data in the mid-1990s, for example, in corpora developed in EU-funded projects such as PAROLE, data used in the US-DARPA Message Understanding Conferences (MUC) [8], and the TIPSTER annotation architecture [10] defined for the NIST Text Retrieval Con-

[1] http://nlp.stanford.edu/software/tagger.shtml

[2] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

[3] http://www.coli.uni-saarland.de/ thorsten/tnt/

[4] Several initiatives have focused on reusability of language data from the late 1980s onward; see Chapter IV in this volume for a fuller history of standards efforts in the field.

[5] Note that the Hypertext Markup Language (HTML) is an *application* of SGML/XML, in that it uses the SGML/XML meta-format to define specific tag names and document structure for use in creating web pages.

[6] www.tei-c.org/

[7] http://www.ilc.cnr.it/EAGLES/browse.html

ferences (TREC)[8], which included a CES-based SGML format for exporting output from information extraction tasks. SGML and XML were also adopted by major annotation frameworks developed during this period, such as GATE[9] and NITE[10], for import and export of data.

Although widely adopted, XML as an in-line format for representing linguistic annotations did not solve the reusability problem, for several reasons. First and foremost, XML requires that in-line tags are structured as a well-formed tree, thus disallowing annotations that form overlapping hierarchies and making connections between discontiguous portions of the data cumbersome. In addition, like all in-line formats, the insertion of of annotation information directly into the data imposes linguistic interpretations that may not be desired by other users. This includes segmental information–e.g., delineation of token boundaries in-line, whether by surrounding a string of characters with XML tags or separating it with white space, line breaks, or other special characters–as well as the inclusion of specific annotation labels and features. To solve this problem, in 1994 the notion of *stand-off annotation* was introduced in the CES[11], wherein annotations are maintained in separate documents and linked to appropriate regions of primary data, rather than interspersed in the primary data or otherwise modifying it to reflect the results of processing. This allows different annotations for the same phenomenon to co-exist, including variant segmentations (e.g. tokenizations) as well alternative analyses produced by different processors and/or using different annotation labels and features.

Annotation Graphs (AG) [21], introduced in 2001, are a standoff format that represents annotations as labels on edges of multiple independent graphs defined over text regions in a document. Because the model was developed primarily with speech data in mind, the regions are typically defined between points on a timeline, although this is not necessary. However, because each annotation type or layer is represented using a separate graph, the AG format is not well-suited to representing hierarchically-based phenomena such as syntactic constituency.[12]

Over the past decade, there has been an increasing convergence of practice for representing linguistic annotations in the field, with the aim of ensuring maximal reusability but also reflecting advances in our understanding of means to best structure and organize data, especially linked data intended for access and query over the web. In addition to the use of stand-off rather than in-line annotations, focus has shifted from identifying a single, universal format to defining an underlying data model for annotations that can enable trivial, one-to-one mappings among representation formats without loss of information. The most generalized implementation of this approach is the International Standards Organization (ISO) 24612 Linguistic Annotation Framework (LAF) [14, 20] (see also Section 5), which was developed over the past decade to provide a comprehensive and general model for representing

[8] http://www-nlpir.nist.gov/related_projects/tipster/trec.htm

[9] http://gate.ac.uk

[10] http://groups.inf.ed.ac.uk/nxt/index.shtml

[11] Originally called "remote markup"–see http://www.cs.vassar.edu/CES/CES1-5.html#ToCOview

[12] An *ad hoc* mechanism to connect annotations on different graphs was later introduced into the AG model to accommodate hierarchical relations.

linguistic annotations. To accomplish this, LAF was designed to capture the general principles and practices of both existing and foreseen linguistic annotations, including annotations of all media types such as text, audio, video, image, etc., in order to allow for variation in annotation schemes while at the same time enabling comparison and evaluation, merging of different annotations, and development of common tools for creating and using annotated data.

Early in its development, LAF defined a set of fundamental architectural principles, including the clear separation of primary data from annotations, and separation of annotation structure (i.e., physical format) and annotation content (the categories or labels used in an annotation scheme to describe linguistic phenomena), and a requirement all annotation information be explicitly represented rather than building knowledge about the function of separators, position, etc. into processing software. It also defined an abstract data model for annotations, consisting of an acyclic digraph decorated with feature structures, grounded in *n*-dimensional regions of primary data. The LAF data model and architectural principles, which in large part simply brought together existing best practices from a variety of sources, significantly influenced subsequent development of models and strategies to render linguistic annotations maximally interoperable. As a result, most general-purpose representation formats developed over the past decade embody most if not all of LAF's principles. Formats to enable interoperability within large systems and frameworks have also followed many of the same principles and practices, for example, the Unstructured Information Management Architecture's (UIMA) [24] Common Analysis System (CAS). The convergence of practice around the graph-based data model has led to the realization of increased compatibility of formats via mapping, and, as a result, transducers among formats are increasingly available that allow for the processing of annotated language resources by different tools and for different purposes (e.g., ANC2Go [22], Pepper [15], and transducers available with DKPro[13]).

There remains, however, a tension between ease of processing and meeting the demands of interoperability. Along with the more verbose and complex formats described above and in some of the following sections, *column-based* representations have gained increasing usage. The most well-known of these is the CoNLL IOB format, which was designed several years ago for use in the Conference on Natural Language Learning[14] shared tasks. The format was devised to allow for multiple annotations over the same data and to be easily machine-processable by diverse teams and software, and the format's simplicity, ease of processing, and human readability have made CoNLL a popular format despite the awkwardness of representing certain types of information (e.g., syntactic hierarchies). At the same time, due to its popularity, transducers to and from CoNLL format for some general-purpose formats exist, and as a result, finding a format that is both amenable to in-house processing and readily importable from and exportable to any of several formats is increasingly achievable.

[13] http://www.ukp.tu-darmstadt.de/research/current-projects/dkpro/

[14] http://ifarm.nl/signll/conll/

Column-based formats such as CoNLL can be considered a hybrid form of stand-off markup, in that they do not annotate primary data but rather annotate a segmental annotation of the primary data, in particular, tokens extracted from the primary data, listed one per line. Other "hybrid-standoff" approaches utilize XML inter-document reference mechanisms such as XPointer and Xlink to associate annotations to XML elements embedded in primary data (e.g., [37]). Hybrid approaches have the disadvantage of imposing a layer of linguistic interpretation (e.g., what constitutes a token, sentence, etc.) that may not be desired by other users. In addition, the "one token per line" assumption adopted in the CoNLL format can seriously handicap algorithm performance: for example, some phenomena (e.g., hashtags in tweets) need to be split apart as separate tokens in order to assign part-of-speech tags to the constituents, but the requirement that individual tokens must appear on separate line loses the information that the constituents appear as a unit in the text. However, despite their limitations, hybrid approaches offer certain advantages for processing ease and, in the case of XML, readily available tool support.

3 Representation Schemes for Text

3.1 Segments

The problem of representing linguistic annotations for textual data invariably starts with the decision on the minimal unit of the analysis, i.e., the smallest portion of the text that may receive an annotation. Very often, the minimal unit in textual data is the word or *token*, although in some cases the minimal units may be smaller (e.g., morphological units) or larger (e.g., sentences). Regardless of the size and nature of the minimal unit of analysis, its identification reflects a decision or viewpoint that may be based on linguistic, processing, or task-dependent grounds. As such, identification of the minimal units of analysis can be regarded as a first-level *annotation* of primary data that imposes an interpretation of its characteristics, which may vary from project to project.

Once the minimal unit of analysis is determined, the next step is to *segment* the text–i.e., to identify *continuous* spans of text that are unambiguously identifiable via automatic means, and which provide the pieces of the data that will be used to make up the minimal units of analysis. Often, the segments are identical to the minimal units of analysis, although in some cases, the segmentation may identify spans smaller than the minimal unit, especially when the minimal unit may consist of discontiguous spans of text.[15] The segmentation may or may not cover the whole of the data or be continuous; a segmentation into tokens or sentences, for example, will often cover the entire text (ignoring white space), but, although less common,

[15] In addition, to solve the well-known problem of representing alternative tokenizations over the same data is well-known, segmentation into smaller units that may be combined to form differing tokenizations has been proposed [38, 19].

a segmentation may also isolate only higher-level phenomena, e.g., noun chunks or named entities, and thus cover only certain portions of the text.

How the text as a sequence of segments is physically represented depends on the choice of format. We can roughly distinguish four basic approaches:

Inline linear formats. In simple plain-text inline representations, often found in older corpora, segments (most commonly, tokens) are white-space separated, and annotations may be "attached" to each segment with a special character (e.g., vertical bar, underscore, slash).

Inline XML. Segments in an XML document are represented by surrounding each relevant span with an XML tag, typically including an *id* attribute that can be referenced from other annotations. In some cases, attributes providing additional annotation information (e.g. for tokens, attributes such as part-of-speech tag, lemma, etc.) are also included. In other cases, the XML document is treated as a base for other annotations that are contained in separate XML documents that reference base segments via their *id* attribute values.

Column-based formats. These representations extract segments that serve as the minimal unit of analysis (again, usually tokens) from the primary data, at which point the primary data and any information about the location of the segment in the text are effectively discarded. A new document is produced in which each minimal unit appears on a new line, and annotations for that unit can be added to the line, each separated by a special character.

Standoff annotations. In a stand-off representation, segment boundaries are not indicated in the primary data document, which is treated as "read-only"; rather, segments are identified in a separate document that specifies the start and end offsets of each segment in the primary data document.

3.2 Annotation structure

Besides defining the units that receive annotation, the second essential representation decision concerns the structure of the annotations. Depending on the task, annotations can be single labels, sets of "flat" attribute-value pairs, full-fledged recursive feature structures, relations between segments, or various combinations of these. In any of these cases, a representation format for the annotation information itself must be determined, which, for more complex annotation structures such as features structures, can be a non-trivial task. In addition, it is necessary to identify the *pairing* of segments-to-be-annotated and the annotated information in some way. For typical annotation scenarios, we can roughly distinguish four cases, which are discussed in the following subsections.

3.2.1 Inline annotation of plain text

Inline annotations add linguistic information directly to the segmented text. In plain text representations, the most straightforward scenario involves attaching a single label to a single base segment: a case in point is part-of-speech (POS) information, which in a linear format can be represented, for example, as a sequence of token/annotation pairs, for example: `Many_DET cultural_ADV treasures_N).` Another prototypical scenario for inline segment labeling is syntactic chunking, where the text is interleaved with labels for categories such as *noun chunk*, *verb chunk*, etc. Similarly, named-entity (NE) annotation may associate token sequences with information that identifies and characterizes an entity such as a person, location, etc. For example, in the following, square brackets delimit segments annotated with NE types in capital letters:

[FACILITY Many cultural treasures] are, however, not in a representative state. [GROUP We] have to restore [FACILITY them].

The most well known example of inline segment labeling is the format of the Penn Treebank, which utilizes nested bracketing to represent the structure of a constituency parse and intersperses both part of speech and constituency labels within the text:

```
( (S (NP-SBJ (NNP Bartok))
     (VP (VBZ describes)
                (NP (NP (DT the) (NN form))
            (PP (IN of)
                      (NP (DT the) (JJ first) (NN movement))))
            (PP-CLR (IN as)
                     (NP (NP (ADJP (ADVP (" ") (RBR more)
                                 (CC or) (RBR less)) (JJ regular))
                                       (NN sonata) (NN form)))))
            (, .)
```

Inline annotations are straightforward and easy for humans to read, and formats such as those shown above were widely used from the 1960s throughout the early 1990s (e.g., the Brown Corpus). However, data in this form are notoriously difficult to modify or add to, and generally require specialized software to process, and as a result, inline formats of this kind are rarely used today. In addition, they pose problems for handling discontiguous segments, as discussed below.

3.2.2 Inline XML

In general, using XML has the advantage of a solid base of supporting technology to create, validate, and process XML documents. When annotations are represented with standard inline XML, XML elements are used to mark the beginning and end of a segment and/or a contiguous group of segments of which it is comprised. For example, the sentence above could be represented in XML as follows:

```
<S><FACILITY>Many cultural treasures</FACILITY> are, however,
not in a representative state.
<GROUP>We</GROUP> have to restore <FACILITY>them</FACILITY>.</S>
```

Note that the same example could be represented in a variety of ways, since XML only provides the syntax of tag use and does not define a standard set of elements, or even dictate what is an element name and what is an attribute (the FACILITY element in the above might be rendered as `<ENTITY type="facility">`, for example). A classic example of an inline XML representation is the British National Corpus[16], which uses the Text Encoding Initiative (TEI) XML Guidelines to annotate the data with part of speech tags and for logical structure (paragraph, heading, etc.). However, for more complex kinds of annotation, complications arise when segments overlap, since the inherent hierarchical structure of an XML document is violated. Various solutions are available (e.g., the use of *milestones* to mark segment boundaries), but the "spirit" of an XML document is then lost and, more importantly, many XML tools cannot process such documents. Other problems arise when segments are *discontiguous*, which can happen for instance in the annotation of referring expressions, or when relative clauses are to be treated as forming a single unit with their head NP. In a language like German, the two need not be adjacent:

[Ich]$_{ref.1}$ habe [einen Hund]$_{ref.2}$ gesehen , [der sehr alt war]$_{ref.2}$.
I have a dog seen , that very old was .
'[I]$_{ref.1}$ have seen [a dog that was very old]$_{ref.2}$.'

To represent discontiguous elements in an inline XML representation, some form of co-indexing is required to relate the parts of the referring expression to one another; this is typically accomplished by giving a common ID to the tokens that combine into a segment, as suggested by the example above.

3.2.3 Column-based annotations

As noted above, column-based formats extract the text segments that will serve as the minimal units of analysis from the primary data and create a new document that serves as the basis for the annotations. The annotations for each minimal unit (here, we consider that to be the token) are given on the same line as the token. When an annotation spans several contiguous tokens, the common strategy is to use the "BIO" format; for example, in the following,

```
0    Many      B-NP
1    cultural  I-NP
2    treasures I-NP
3    are       O-NP
...
```

[16] http://www.natcorp.ox.ac.uk

`B-NP` signals the beginning of a noun phrase, `I-NP` indicates the token is "in" the noun phrase, and `O-NP` says it is outside a noun phrase.

The column-based format has the advantages of ease of processing and readability by humans. Also, it is trivial to represent multiple layers of annotation as well as add new ones, since columns can be added freely. A disadvantage is that the columns need to be interpreted: their role is not made explicit in the representation as it is, for example, in the element names and attributes of an XML format, and users of the format need to agree on what information goes where.

The column-based format also has the disadvantages of imposing fixed base segmentation and losing much orthographic and presentational information from the original text. Perhaps most seriously, it does not readily handle hierarchical annotations (e.g., syntax trees) or annotation of discontiguous tokens. As with inline XML, co-indexing is required to specify hierarchical relations or relate discontiguous items, and such co-indexing substantially complicates the processing of documents in this format.

3.2.4 Standoff and "hybrid standoff" annotations

Many annotation projects annotate multiple linguistic layers, from tokenization and morphosyntax to syntax and beyond. The multi-layer scenario corresponds to the notion of *tiers* used in common approaches to speech annotation–see Section 4.2. However, as the kinds and number of annotations increase, representing them in a way that enables them to be used and processed together becomes more and more complicated. The common approach to multi-layer annotation therefore is to use standoff annotation, which allows for a clean separation of the primary data (text) and the various annotation layers.

In its purest form, standoff annotation is applied to a frozen, read-only version of the primary data, and all segmentations and annotations are provided in separate documents that reference offsets in the data (or other annotations–see below). The intent is to retain all information in the original text for possible future reference; corrections or normalizations of the data are handled as annotations themselves. A "hybrid-standoff" approach creates a new document from primary data containing the basic segments. One common hybrid-standoff strategy represents the segments with inline XML and uses XML inter-document reference mechanisms such as XPointer and Xlink to associate annotations to the XML elements embedded in that document (e.g., PAULA/XML, described in Section 5). Column-based formats such as CoNLL can also be considered a hybrid form of stand-off markup, in that they do not annotate primary data but rather a segmental annotation of the primary data.[17] Hybrid approaches have the disadvantage of imposing a layer of linguistic interpretation (e.g., what constitutes a token, sentence, or a syntactic constituent)

[17] An extreme example of hybrid standoff is the format used in PropBank[18], which uses a form of Gorn addressing to attach semantic role annotations to nodes in the syntax trees defined in the original Penn Treebank.

that may not be desired by other users; at the same time, they offer certain advantages for processing ease and, in the case of XML, readily available tool support.

In either form, standoff annotation readily handles hierarchical structures, discontiguous segments, and intra- and inter-document references because it can simply reference the locations of the segments to be annotated in either primary data or the document containing base segments. In a multi-layer scenario, to associate annotations with the data, three possibilities exist: each layer can point directly into primary data as, for example, in the strategy originally proposed for Annotation Graphs, where every annotation regardless of layer directly references spans in the primary data; annotations can reference *only* the minimal units identified in the document containing the base segments; or annotations can reference minimal units and/or annotations in others layers of analysis (e.g., a named entity annotation can reference its component tokens, or a Sentence annotation can reference annotations for its constituent NPs and VPs). For text, the third strategy is the preferred method for multi-layered annotations.

As an example, consider the following representation of a token annotation of "three-fold" in LAF/GrAF. Three segments (regions) are defined via *anchors* that point into read-only primary data using 0-based offsets. A node in the annotation graph links to the three segments, thus associating them *as a unit* with a token annotation that includes features for part of speech (msd).[19] The segments in this case happen to be contiguous, but that is not required.

```
<region xml:id="seg-r770" anchors="211 216"/> <!-- "three" -->
<region xml:id="seg-r771" anchors="216 217"/> <!-- "-" -->
<region xml:id="seg-r772" anchors="217 221"/> <!-- "fold" -->

<node xml:id="n1019">
        <link targets="seg-r770 seg-r771 seg-r772"/>
</node>
<a label="tok" ref="n1019" as="xces">
        <fs>
            <f name="msd" value="JJ"/>
        </fs>
</a>
```

Fig. 1 Referencing segments in GrAF

Other annotations can be linked to one or more token or other annotation in the graph by defining an edge from their associated nodes to the node or nodes to be annotated using the node element ids, rather than pointing directly into the primary data. Thus annotations can be built up as a directed acyclic graph over the primary data, with the primary data segments serving as terminals.

The MMAX2 annotation tool [60] uses a hybrid standoff representation that is defined over an inline XML segmentation into tokens. In the MMAX2 vernacular,

[19] In GrAF, each annotation is "attached" to a node in the annotation graph.

elements that can be annotated are called *markables*, and they can be represented by pointing to to a single token, or a span of contiguous tokens, or discontiguous token spans. Markables receive a unique ID; annotations are added to them as XML attribute/value pairs. Figure 2 shows an example from an annotation layer for referring expressions.

```
<markable id="markable_74" span="word_141..word_142"
   grammatical_role="sbj" referentiality="discourse_new" ...
<markable id="markable_1000151" span="word_151"
   grammatical_role="sbj" anaphor_type="anaphor_nominal" ...
<markable id="markable_1000153" span="word_153"
   grammatical_role="dir-obj" anaphor_type="anaphor_nominal" ...
```

Fig. 2 Fragment from an MMAX layer for referring expressions

Multiple layers can independently define their markables by referring to tokens, or (in the case of the MMAX2 model) to other markables in other layers. Thus, an annotation layer can provide information either about primary data segments or other annotations.[20]

Another example of hybrid approach is the GATE annotation model, which is based on Annotation Graphs [69]. It inserts zero length "node" annotations into the original document content that serve as annotation anchors, which allows for different segmentation-based annotations of the same type (e.g., different tokenizations) to be represented simultaneously. The representation maps directly to a fully standoff XML representation, where all annotation layers are linked to the nodes in the original text. The disadvantage of this approach is that relationships among different annotation layers (e.g., shared or overlapping spans) cannot be represented.

3.3 Relation annotation

Some annotation types require the annotation of *relations* between segments (or between annotations of segments). A clear example is dependency syntax, where functional relations are introduced between words in the sentence; these relations are directed and point to "heads".

Relational annotations may be *directed* or *undirected*. For example, nominal coreference, signifying that two NPs refer to the same entity in the world, can be represented as an *undirected* relation, as can relational annotations for parallel text

[20] Note that the decision to represent annotation layers in this fashion does not automatically lead to the distribution of layers across separate data files. While the MMAX2 model and others (see Section 5) indeed use one file per layer, other approaches such as that of the model underlying the Serengeti tool [30] prefer combining all information into a single file, which begins with the token layer and then lists the various standoff annotation layers.

alignment, i.e., linking the corresponding words or sentences of the same text in different languages. Anaphoric coreference, on the other hand, is represented as a *directed* relation from the anaphoric NP (often, a pronoun) to its antecedent. Similarly, temporal relations (as in TimeML–see Part II, III.f) that link events according to their relationships over time are typically not only directed, but also annotated to specify their type (e.g., "before", "after", etc.).

MMAX2 allows for two types of relations:

1. Undirected relation: An arbitrary number of markables can be linked together, thus establishing a *set* of markables.
2. Directed relation: Given a "source" markable M and one or more "target" markables $T_1, T_2, ..., T_n$, pointers can be established from M to the T_i.

In GrAF, relations are typically represented as edges between nodes. All edges in GrAF are by default directed, but edges as well as nodes may be labeled with annotation information. Thus, for example, an undirected edge between nominal coreferents could be annotated with the label *nom-coref* and have a feature that gives its type as "undirected".

3.4 Hierarchical structures

Hierarchical structures are common in syntactic analyses. When individual dependency relations combine to a full analysis of a sentence, they encode a hierarchical structure, but one that does not require "extra" nodes beyond the words (i.e., the words themselves serve as nodes in the graph or tree). In contrast, constituency syntax trees require extra nodes to represent the constituents, which are themselves annotations of the primary data or other constituents (annotations) for a given sentence. Other annotation scenarios also involve tree structures; for example, Rhetorical Structure Theory [33] posits that the structure of complete texts can be modeled as trees. Here, we use constituency syntax as the prime example.

As noted in Section 2, the first major syntax treebank, the Penn Treebank [3], was distributed as a set of plain text files with syntax trees encoded via brackets and indentation, following the conventions of the Lisp programming language. Later on, column-based formats were devised for this purpose, an early instantiation being the "NEGRA export format", developed as part of the first German syntax treebank NEGRA [35]. The column-based format is also popular for representing dependency parses, in which each word is given a unique ID, and, after the POS and morphology columns, the following information is specified in individual columns: a pointer to (the ID of) the associated head token; the dependency relation to this head; the ID of a projective head; and the associated dependency relation.

In contrast to the column-based representation of dependency trees, NEGRA requires the addition of extra lines that do not represent a word of the text, but rather a syntactic constituent. The convention is, for each sentence, to first give the sequence of word lines and then, in no particular order, a set of constituent-representing lines.

```
#FORMAT 3
#BOT ORIGIN
1                       refcorpus %% Stuttgarter Referenzkorpus, Frankfurter Rundschau
#EOT ORIGIN
#BOT WORDTAG
1                       skup          Wojciech
#EOT EDITOR
#BOT WORDTAG
-1                      UNKNOWN       N                   Unbekanntes Tag, Fehler
0                       --            N                   nicht zugeordnet
1                       ADJD          Y                   Attributives Adjektiv
2                       KOUS          Y                   Unterordnende Konjunktion mit Satz
3                       NN            Y                   Normales Nomen
4                       PIAT          Y                   Attribuierendes Indefinitpronomen
5                       PRELS         Y                   Substituierendes Relativpronomen
6                       VAFIN         Y                   Finites Verb, aux
8                       VVFIN         Y                   Finites Verb, voll
9                       $,            N                   Komma
10                      $.            N                   Satzbeendende Interpunktion
#EOT WORDTAG
#BOT MORPHTAG
-1                      UNKNOWN       unknown tag, error
0                       --            not bound
1                       3.Akk.Pl      3rd person, accusative, plural
2                       3.Sg.Pres.Ind 3rd person, singular, present, indicative
3                       Masc.Nom.Sg   masculinum, nominative, singular
4                       Masc.Nom.Sg.* masculinum, nominative, singular, *
5                       Pos           positive
6                       *.*.*.*       underspecified
#EOT MORPHTAG
#BOT NODETAG
-1                      UNKNOWN       unknown tag, error
1                       NP            noun phrase
0                       --            not bound
2                       S             sentence
#EOT NODETAG
#BOT EDGETAG
-1                      UNKNOWN       unknown tag, error
1                       NP            noun phrase
1                       CP            complementizer
2                       HD            head
3                       NK            noun kernel modifier
4                       OA            accusative object
5                       PD            predicative
6                       RC            relative clause
7                       SB            subject
#EOT EDGETAG
#BOT SECEDGETAG
%% no secondary edges used
#EOT SECEDGETAG
#BOS 12 1 847184076 1
Shade                   ADJD          Pos                 PD              503
,                       $,            --                  --              0
daß                     KOUS          --                  CP              502
kein                    PIAT          Masc.Nom.Sg.*       NK              501
Artz                    NN            Masc.Nom.Sg.*       NK              501
anwesend                ADJD          Pos                 PD              502
ist                     VAFIN         3.Sg.Pres.Ind       HD              502
,                       $,            --                  --              0
der                     PRELS         Masc.Nom.Sg         SB              500
sich                    PRF           3.Akk.Pl            OA              500
auskennt                VVFIN         3.Sg.Pres.Ind       HD              500
.                       $.            --                  --              0
#500                    S             3.Sg.Pres.Ind       RC              501
#501                    NP            Masc.Nom.Sg.*       SB              502
#502                    S             3.Sg.Pres.Ind       --              503
#503                    S             *.*.*.*             --              0
```

Fig. 3 Example NEGRA dependency parse representation

To preserve compatibility with the columns on the word lines, NEGRA uses a similar layout and fills the non-applicable columns with "ZERO". Thus a constituent line consists of: ID; ZERO (no equivalent to lemma); syntactic label; ZERO (no equivalent to morphology); grammatical function; ID of mother node. To provide the link between words and constituents, the final columns of word lines also encode the ID of the mother constituent node; optionally this can be followed by the label of a secondary edge (see below), and by the ID of its target node. Figure 3 shows the

sentence “Schade, daß kein Artz anwesend ist, der sich auskennt” represented using the NEGRA format.

The syntactic structure in NEGRA (and in the follow-up project TIGER, see Part II, II.b) is by definition relatively flat, and both schemes use the instrument of “secondary edges” to encode long-distance dependencies. Since they lead to crossing edges, they violate the constraints of trees; for this reason, the annotations cannot be represented by simple bracketing of the source text and inserting constituent labels, as in the PTB.

For the same reason, the embedding structure of XML documents cannot adequately capture syntactic representations in the style of TIGER. In that project, a specialized XML-based exchange format was designed to supplement the column format in which the hierarchical structure of the XML elements in the document was not used to represent relations among constituents. Instead, TIGER XML [34] encodes the hierarchy information with pointers: mother nodes point to daughters with the IDREF attribute. Both nodes and edges are XML elements, so that edges, too, can be labeled. Similar to the column format, the XML format first lists the terminal nodes (tokens) with lemma, POS, and morphology information; then nonterminals are described by ID, category, and a list of edges with labels and pointers to target IDs. For illustration, Figure 4 shows the representation of a German example sentence in TIGER XML.

4 Representation Schemes for Multi-modal Data

Multimodal data is presented here as an alternate to purely textual data. It generally includes digitised audio and video recordings but can also refer to time-based signals recorded from various physiological or environmental observations. The defining feature of multimodal data is that it is time based and that in its digital form: it is represented as a sequence of samples in a digital signal. A digitised signal is defined in part by a *sample rate* which is the number of times per second that the value of the signal is recorded. The sample rate defines the maximum resolution of any annotation on the signal – it is not possible to observe, and therefore annotate, any phenomenon that occurs between two samples of the signal.

While ‘multimodal’ refers explicitly to more than one mode (of communication), it is often used to refer to single-mode recordings of audio or video data. True multimodal data would consist of more than one modality. When there is more than one signal then there is often more than one sample rate (e.g. 44100Hz for audio, 30Hz for video) and so the *alignment* of signals and the annotations on the signals becomes an issue. Having said this, the models of annotation used for one or many signals are largely the same but different annotation tools support different kinds of source data.

```
<s id="s28" art_id="1">
  <terminals>
    <t id="s28_1" word="Viele" lemma="--" pos="PIAT"
        morph="--"/>
    <t id="s28_2" word="Kulturschatze" lemma="--" pos="NN"
        morph="--"/>
    <t id="s28_3" word="sind" lemma="--" pos="VAFIN"
        morph="--"/>
    <t id="s28_4" word="aber" lemma="--" pos="ADV" morph="--"/>
    <t id="s28_5" word="nicht" lemma="--" pos="PTKNEG"
        morph="--"/>
    <t id="s28_6" word="in" lemma="--" pos="APPR" morph="--"/>
    <t id="s28_7" word="einem" lemma="--" pos="ART" morph="--"/>
    <t id="s28_8" word="prasentablen" lemma="--" pos="ADJA"
        morph="--"/>
    <t id="s28_9" word="Zustand" lemma="--" pos="NN"
        morph="--"/>
    <t id="s28_10" word="." lemma="--" pos="\$." morph="--"/>
  <terminals>
  <nonterminals>
    <nt id="s28_500" cat="NP">
       <edge label="NK" idref="s28_1"/>
       <edge label="NK" idref="s28_2"/>
    </nt>
    <nt id="s28_501" cat="PP">
       <edge label="AC" idref="s28_6"/>
       <edge label="NK" idref="s28_7"/>
       <edge label="NK" idref="s28_8"/>
        <edge label="NK" idref="s28_9"/>
    </nt>
    <nt id="s28_502" cat="S">
        <edge label="SB" idref="s28_500"/>
        <edge label="HD" idref="s28_3"/>
        <edge label="MO" idref="s28_4"/>
        <edge label="NG" idref="s28_5"/>
        <edge label="MO" idref="s28_501"/>
    </nt>
    <nt id="s28_VROOT" cat="VROOT">
        <edge label="--" idref="s28_502"/>
        <edge label="--" idref="s28_10"/>
    </nt>
  </nonterminals>
  </graph>
  </s>
```

Fig. 4 German example sentence in TIGER XML

4.1 Varieties Multimodal Annotation

Multimodal data is used by a range of disciplines and consequently there are a number of different styles of annotation that are used. Schmidt et al [68] provides a useful summary from the point of view of the *use* of the corpora. Here we will characterise the range of annotation styles based on the formal structure and representation of the annotations.

4.1.1 Transcriptions of Speech

Many researchers interested in speech are mainly concerned with the language that is used rather than the acoustics of the underlying speech signal. In such cases it is common to use transcripts of spoken recordings that either have no time-based reference to the original recording or where the time references are at a very coarse-grained level. Schmidt et al refer to these as *spoken language corpora* and they are widely used in linguistic research where the focus of interest is at the lexical level and above.

It is common for transcriptions to be done using tools commonly used for transcribing meetings or court proceedings etc; that is, the speech is transcribed into a word-processor with speaker turns marked in the style of a movie script. Figure 5 shows a small excerpt from this kind of transcription that illustrates the use of speaker turn labels and some embedded markup - in this case, square brackets indicating overlap between the two speakers.

```
RF3:  Okay.  And what about your immediate family?
T3M:  Yeah, I've got one sister and well the dog he's part of the
      [family so yeah]
RF3:  [Of course.]  Is your sister older or younger?
T3M:  She's younger.  She's uh gunna turn eleven in July.
RF3:  Oh I see.
T3M:  Yeah.
RF3:  So what grade's she in?
T3M:  She's in Year Five at the moment.
RF3:  Right.
```

Fig. 5 An example of a transcribed spoken recording taken from the Monash Corpus of Spoken English [67]

In some cases timestamps are included, often aligned with the start of each speaker turn but in some cases just 'every now and then'. The purpose of the timestamps is usually to allow a researcher to return to the audio recording to manually listen to a region of speech in case the transcript is ambiguous or unclear. As a result, the timestamps don't need to be too accurate and are often expressed to the nearest second.

This style of data is often treated as a textual data source once the transcription has been carried out – with no further reference made to the original recording. Hence annotations on transcripts can be thought of as a kind of textual annotation and all of the prior discussion in this chapter is relevant.

One widespread and well developed example of this style of annotation is that generated by the CLAN tools developed for the CHILDES/Talkbank project.[21] These tools support the creation of a sophisticated style of transcription that can be aligned with an audio or video recording. CLAN transcripts can range from simple transcripts to multi-layered analyses of conversation and the toolkit supports a range of transformations and analysis methods on the data as well.

Another widely used style of transcription is *Conversation Analysis* [76] which adds a collection of annotation markers to a transcribed turn-by-turn conversation to denote various non-lexical phenomena such as pauses, overlapping speech, changes in pitch, etc. While there is some agreement on the characters used to mark these different phenomana, there is generally no way to enforce a particular style as these analyses are usually carried out using a general purpose word processor.

4.1.2 Interlinear Text

Interlinear Text (IT) is a style of transcription of spoken language widely used in Linguistic fieldwork to record utterances in a language under study along with some analysis and a *gloss* or loose translation into another language. While it is widely used as a purely written form of transcription, there is increasing interest in developing Interlinear Texts that are time aligned with an audio recording.

Here is an example interlinear text that describes the analysis of an utterance in Classical Nahuatl:[22]

```
ni- c- chihui  -lia in     no- piltzin ce calli
I   it make    for  to-the my  son     a  house
I made my son a house.
```

The first line of the analysis is a transliteration of the spoken form split into words by spaces and into morphemes by hyphens. Below this is an English gloss for each morpheme and below that an English translation of the sentence as a whole. The vertical alignment of the parts of the analysis is what characterises this as an Interlinear Text. While this example does not include any temporal information, it is now common to build this kind of analysis using tools such as ELAN[23] which support anchoring one or all of these tiers into a timeline.

Interlinear Text is often discussed as a special mode of annotation, for example Bow et al [70] present a review of the many styles of IT and then develop an abstract model of IT as annotations. However, it can be usefully seen as just a way of

[21] http://childes.psy.cmu.edu

[22] Taken from http://www.ling.hawaii.edu/ldtc/website/syllabus/sp06/LehmannGlossing.pdf

[23] https://tla.mpi.nl/tools/tla-tools/elan/

visualising a class of aligned annotations; Schmidt [79] develops a model of *IT as visualisation* that usefully characterises the kinds of annotations that can be treated in this way.

4.1.3 Acoustic Segmentation

In the most common style of annotation on multimodal data, the temporal signal is segmented into discrete chunks which are then labelled with one or more simple textual labels. Different kinds of annotation can be made on the same signal and these are organised into layers or *tiers* containing all of the annotations of a particlar type. These annotations are generally made using special software applications that allow visualisation of the speech signal and derived signals such as a spectrogram or pitch track, although, in some cases, automated annotation is carried out using adapted speech recognition software.

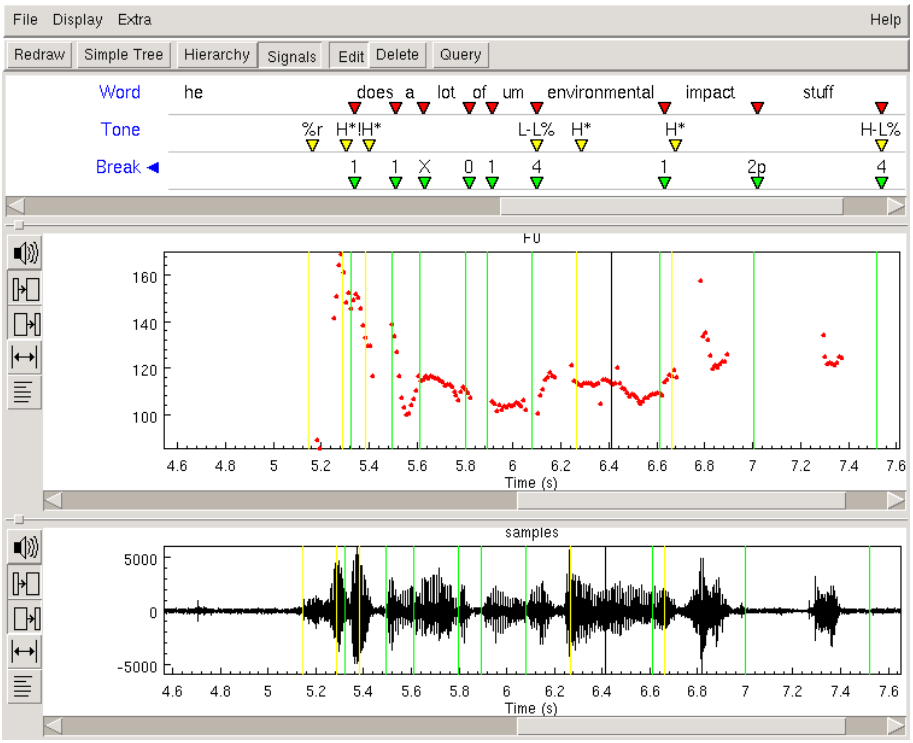

Fig. 6 An example acoustic phonetic annotation using the ToBI scheme showing word segments, Tone events marking locations in the pitch track and Break events marking the perceived degree of juncture on breaks between words [81].

As an example, Figure 6 shows a speech waveform and associated pitch trace that is being annotated using the Emu labeller[24] according to the ToBI [81] guidelines. The upper panel in the figure shows the annotation displayed in the typical *musical score* style with the time locations marked by small triangles. The annotations are shown in three tiers where the word tier contains segments with a start and end time and the Tone and Break tiers contain events with just a single time for each annotation.

This style of annotation is used for different levels of analysis from fine-grained phonetic segmentation to larger chunks like syllables, morphemes and words. In many cases different tiers are used to combine many different levels of analysis on the same signal. Some tools support the creation of links between the segments in different tiers to support a hierarchical analysis of the signal. For example, words may contain syllables which contain phonemic segments. Where this kind of linking is not supported, it is common to create implicit links by making the start and end of the dominating segment align with those of the subordinate segments.

4.1.4 Gesture Annotation

A variation on the segmentation of multimodal data is used in the analysis of video recordings of human communication. The temporal location for each segment or event is augmented by the description of a region in the video frame. Figure 7 shows an example of this style of annotation viewed in the ELAN annotation tool; in this case, temporal regions have been marked by an automated annotation tool which finds features such as hand or head movement and joined hands [82].

4.2 Characteristics of Multimodal Annotations

Multimodal annotation is by necessity represented as *standoff* annotation in that annotations are recorded separately to the primary signal being annotated. Beyond that common feature, there are a wide variety of styles of annotation and annotation file formats that are used in the different disciplines that make use of multimodal data.

At the core of all multimodal annotation is the idea of a segment or event in the time stream. Segments are characterised by a start and end time, while events have a single time reference (note that in some cases frame or sample counts might be used in place of time). Segments and events will then either a simple label or a feature structure associated with them.

There are two primary relational structures that are represented in annotations on multimodal data: sequence and hierarchy. Both of these follow from the fundamental structure of speech as both a temporal signal with one sound following the next

[24] http://emu.sourceforge.net

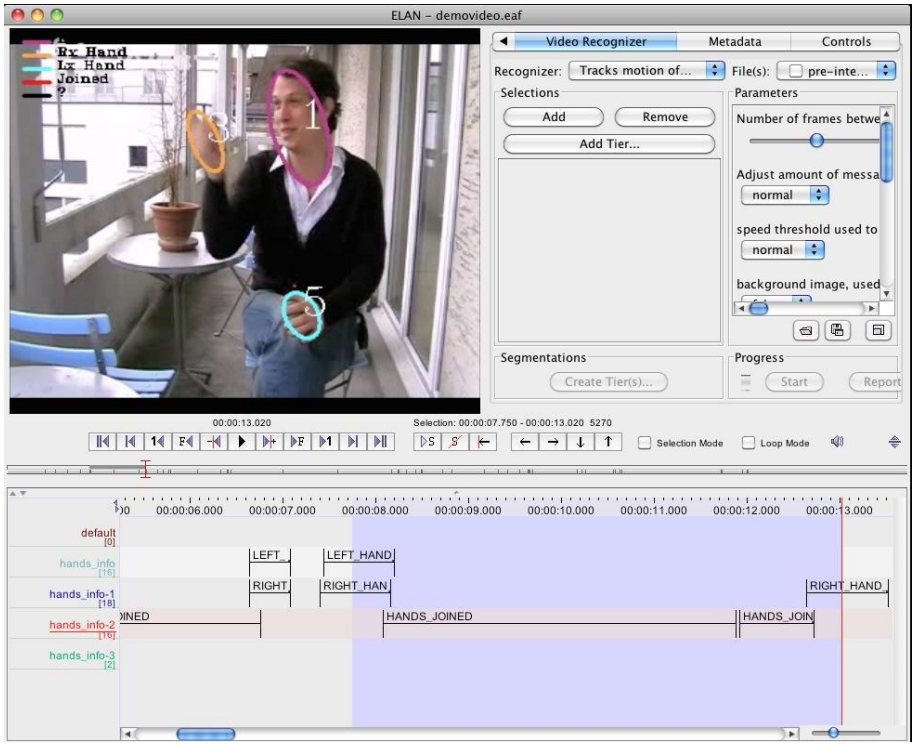

Fig. 7 A screenshot of the ELAN annotation tool with annotations on regions in a video.

and a linguistic structure that can be described on many levels. These structures are reflected in the annotation models that have been developed for multimodal data and most annotation tools implement both of these in some form. Hierarchical relations between annotations usually imply the containment of the children within the parent. In many cases the higher level segments may not have explicit times associated with them since they can be determined by the boundaries of the child segments.

4.2.1 Tiers

Tiers are a common construct in multimodal annotations and most annotation tools support them in some way. A tier is a group of annotations of the same type that have a number of common features; for example, all of the words by a given speaker, or annotations of the left hand activity in sign language. Tiers are used in a number of ways by different tools and in many cases are used as a convenience device to organise annotations and help configure the user interface used to present and edit annotations on a recording. However, tiers are also used as a way of expressing constraints on the annotations on a recording; for example, stating that segments within

a tier must not overlap or that segments on one tier may be in a dominance relation with those in another. In practice, there is some overlap between the concepts expressed by tiers and the idea of *linguistic type* and an *annotation schema* which are realised in some annotation tools.

The simplest version of a tier is a collection of all of the annotations of a given type which are then shown together in an annotation display or authoring tool. This can be seen in the Praat[25] and Emu annotation tools where a tier (Praat) or level (Emu) can be configured as a collection of segments or events with a given type name (Phonetic, Syllable, Word). In both cases there will only be one tier with a given name in the annotations for a single recording.

In other tools, further information can be associated with the tier that applies to all of the annotations it contains. The most common property is the speaker identity with ELAN, Exmaralda[26] and ANVIL[27] supporting this kind of association. A tier is then associated with a given linguistic type and a speaker identifier; this is particularly suited to the annotation of dialogue where each speaker is analysed separately. The use of other tier properties is also possible; for example, ELAN would allow separate tiers for left and right hand annotations of sign language where the type of annotation was the same in each case.

In both cases, tiers are a convenience structure to collect together all annotations with a given combination of properties (type label, speaker identifier, etc) or, to view it another way, as a more compact way to assign common properties to a number of annotations.

Constraints on Tiers: Sequence and Hierarchy

Another use of tiers is to constrain the sequential and hierarchical relations between annotations. In a number of systems, segments within a tier must conform to some constraints such as no-overlap or no-gap segmentation of the time axis. Where hierarchical relations are allowed between annotations, they are often constrained to be between segments in nominated tiers; for example, segments in the Phoneme tier can only have parents in the Word tier.

Constraints are perhaps best illustrated in ELAN which has perhaps the most elaborate set of alternate sequential and hierarchical constraints within and between tiers called the *Linguistic Type Stereotype*:[28]

- None - the 'parent' tier has no restrictions except segments cannot overlap
- Time subdivision - annotation in parent can be subdivided in the child tier with segments linked to time intervals, no time gaps allowed
- Symbolic subdivision - annotation in parent subdivided but no links to time intervals

[25] http://www.fon.hum.uva.nl/praat/

[26] http://www.exmaralda.org/en

[27] http://www.anvil-software.org

[28] www.mpi.nl/corpus/manuals/manual-elan.pdf

- Included in - all annotations fall within parent tier but there can be gaps between segments
- Symbolic association - one-to-one correspondence between parent and child tiers

Sequential constraints within tiers reflect the different semantics of the segments being created. For example, a phonetic segmentation totally sub-divides the speech signal and describes every part of the speech stream and gaps are explicitly represented as segments themselves; in this case, an ELAN *time subdivision* tier would be used and no gaps allowed between segments. On the other hand, a word segmentation might only annotate the start/end point of words in the speech signal which might have gaps between them. The use of these different tier types in multi-modal annotation systems allows some validation of the annotations created and allows an annotation tool to provide an appropriate user interface for creation of annotations.

The hierarchy defined by a tiered structure differs from the kind of hierarchy seen in, say, syntactic annotation which is a true recursive structure with no pre-defined depth. Multi-modal hierarchies are always defined by a fixed set of tiers with pre-defined relations between them. This reflects the kind of phenomena that are encoded in multi-modal annotations, that is, interlinked layered analyses rather than nested hierarchical structures.

In some cases (eg. Praat), the hierarchical relationship between segments in different tiers is left implicit; that is, a segment on the Word tier may span a group of segments on the Phoneme tier but there is no explicit representation of the relationship between them. Praat does provide some user interface convenience shortcuts for aligning boundaries of segments on different tiers to facilitate creating these implicit relationships.

Another distinction in tier types is made in some systems between tiers that refer to the time signal and those that refer to segments in other tiers. In this second kind of tier, the time reference of a segment must be derived from the segments it is related to. For example, in Emu, a Word tier might contain segments that stand in a hierarchical relation to segments in a Phonetic tier; the start and end points of the Word segments will be derived from those of the dominated Phonetic segments rather than being recorded separately for each Word. Similar constructs are used in ELAN and ANVIL. ANVIL also supports tiers (called *sets*) which contain elements with no start/end time that are not linked to another element with a start/end time; these can be used to denote entities that are referenced in a dialogue (eg. a book that is the reference of a pointing gesture).

4.2.2 Timelines

Time is fundamental to the structure of multi-modal annotations and in some annotation systems the idea of a *timeline* is abstracted to allow more flexibility in representing events and segments.

In the simplest case, the start and end times of segments and the times of events are recorded as numerical offsets from some start point: milliseconds, frame number or sample count. The majority of systems record times in this way. However, in some

cases there is a separate representation of a time point that is then used as the start or end of a segment. This further level of abstraction allows a useful extension to the model of segments since the same time-point can be used as the end of one segment and the start of the next - thus the fact that two segments are contiguous is explicitly represented rather than being implicit in their sharing a numerical end and start time. Examples of systems using this kind of representation are Elan, Emu and Exmaralda.

Another function of the abstract timeline is to allow reference to a time-point that doesn't have a time associated with it. For example, in ELAN or Exmaralda one can create a tier containing annotations that sub-divide their parent (eg. morphemic segments within words) but who's times are not made explicit. The ordering of these time points can be referenced and their times are bounded by those of the parent segments, but other than that they are not determined. This is a useful feature that could only be modelled in other systems by forcing an arbitrary time value for each segment (eg. evenly dividing the parent segment); while this can be done, it would tend to imply that the location of each sub-segment has been determined, which it has not. The user would need to be careful in interpreting the annotations.

4.3 File Formats for Multi-modal Tools

Most multimodal annotation is carried out manually using a special purpose application. There are a number of applications designed to cater for different disciplines and styles of annotation. For example, tools that display a waveform and spectrogram (Emu, Praat), those that display video (ELAN, ExMaralda, Anvil), those designed to support transcription of multi-party conversation (Transcriber) etc. There is overlap between tools and researchers will often use more than one tool to create annotations over a set of data. A consequence of this diversity of tools is a corresponding diversity of file formats used to store annotations.

The simplest file format is perhaps that used for the TIMIT corpus;[29] each line contains a start and end time and a label (Figure 8).

```
0 2360 h#
2360 5263 sh
5263 7021 iy
7021 8370 hv
8370 10234 eh
10234 11084 dcl
11084 11462 d
```

Fig. 8 An extract from a TIMIT annotation file contianing the phonetic transcription of the words 'She had'

[29] http://catalog.ldc.upenn.edu/LDC93S1

There are other simple formats that date back to older toolsets, but most modern tools require more information to be stored with the annotation data. This includes grouping the annotations into tiers and recording type information and inter-tier relationships. This has led to a family of more complex file formats. Many of these are based on XML but some (eg. Praat, Emu) are simple text based formats particular to a single tool. While XML is widely used, each tool defines it's own DTD and so file formats are not interoperable.

Fortunately, the commonality between annotation structures is such that it is generally possible to convert one file format into another with little loss of information. Many tools are able to read annotations created by other tools and export annotations into other file formats. Some work has been done by tool authors on defining interoperability standards between tools. A paper by Schmidt et al [80] discusses the issues around interchange of annotations and develops an Annotation Graph based interchange format.

5 Generalized Representation Schemes

As described in the previous sections, there is a variety of options for representing any kind of linguistically-annotated data. Very often, the requirements of in-house or other tools drive the choice of format. However, as annotated data has become more and more available for use by other researchers and tools over the past decade, the need to adapt a particular format for use with other tools, and/or to combine annotations from different sources, of different types, and in different formats has increased. Given the heterogeneity of formalisms involved, it is challenging to integrate their information for either qualitative analysis or NLP applications. This has motivated the development of generalized schemes that abstract away from domain- or tool-specific information, i.e., to be *interoperable*.

The requirements for a generalized format for linguistically-annotated data may extend well beyond those for schemes designed for a specific tool or purpose. In particular, such a format must:

- be capable of representing all linguistic data, including text, speech, audio, video, image, etc, and combinations thereof, as well as the full range of potential annotations over this data, which may be hierarchical and/or relational, refer to discontiguous entities in the data or across other annotations, or reference timelines[30], image regions, video frames, etc.
- provide, via a well-defined underlying model, principled means for transduction to and from other formats
- enable easy and incremental addition, modification, deletion, and merging of annotations, including those from different sources
- aim for maximal processing ease via explicit inclusion of all relevant information, reliance on well-established and readily available processing tools, etc.

[30] See Section 4.2.2

- provide mechanisms for identifying layers, tiers, and other groupings of annotations[31]
- accommodate existing widely-used formats and technologies, such as XML and RDF/OWL
- enable multiple annotations from different sources, e.g. annotations of the same type but using different schemes, etc.
- provide mechanisms for referencing catalogues and repositories of linguistic categories to describe annotations content, and for defining new categories
- provide mechanisms for best practice documentation of the resource

To answer the first requirement, state-of-the-art approaches to corpus interoperability and information integration in multi-layer corpora build on *graph-based data models*. Directed acyclic graphs (DAGs) allow for the representation of all types of linguistic data and annotations, enabling integration as well as means to store and to query all of the annotation information. The graph-based data model is a generalization of models for a wide range of phenomena, including syntax trees, semantic networks, W3Cs Resource Description Framework (RDF)[32], the Unified Modeling Language (UML)[33], entity-relation (ER) models for databases [**?**], etc.–not to mention the overall structure of the web, as a dense inter-connected network of effective objects. It also underlies formats such as the one adopted for internal data exchange in the widely-used UIMA and GATE frameworks. Due to its generality, the graph-based model is both capable of representing any kind of linguistic annotation, whether simple or complex, and enables trivial mappings among formats based on the model. Typically, graph-based annotation formats are primarily intended to serve as "pivot" formats, into and out of which other formats may be mapped for exchange purposes. So, for example, an in-house format can be mapped into and out of the pivot, and therefore, by virtue of similar mappings into and out of the pivot for other compatible formats, achieve mappability and hence interoperability with all of them.

A number of graph-based formats have been proposed over the past decade and a half; one of the earliest is Annotation Graphs [69], which defines multiple independent graphs over primary data, each corresponding to a separate layer or annotation type and consisting of nodes pointing to positions in the data and edges connecting pairs of nodes, with simple labels on the edges containing the annotation information. Later, ISO GrAF [20, 14] defined a format consisting of a *single* graph over primary data, potentially including multiple annotations, consisting of set of nodes, each of which may be decorated with annotation content in the form of a simple or complex feature structure, and a set of directed edges that may also be associated with feature structures providing annotation information (typically, information about temporal, anaphoric, dependency, etc. relations between annotations). Nodes in the graph are associated either with *n*-dimensional *regions* of primary data or with other nodes (annotations) in the graph via directed edges, thus allowing for

[31] See Section 4.2.1.

[32] http://www.w3.org/RDF/

[33] http://www.uml.org

the representation of hierarchical and other relations among annotations. Several similar graph-based formats have been subsequently introduced, some with minor variations (simple labels rather than feature structures for representing annotations, different mechanisms for referencing primary data, etc.), but all are based on the underlying DAG model.

Graph-based models implement a number of general principles and best practices for representing linguistic annotations that have emerged over the past two decades, including the separation of annotation structure (physical format) and annotation content (linguistic information about the data), and the separation of primary data and annotations via support for *standoff annotation*. Unlike many earlier formats and in-line XML, standoff annotation is not embedded in the primary data but rather references regions in it via references to locations in the primary data[34]. This allows for multiple annotations, including multiple annotations of the same type, over the same data, and eliminates the need to "disentangle" annotations from data in order to reuse it for other purposes or with other schemes or tools. For example, with the standoff format different tokenizations of the data can be represented and referenced by annotations from any other level of analysis, several different syntactic analyses can co-exist, etc.

The current state of the art approach to representation of linguistically-annotated data is to use a graph-based representation serialized as standoff XML as a pivot format [71, 78] and relational data bases for querying [50, 53]. Relational databases implement the ER data model, itself aserialization of the graph model, and therefore relational databases are readily created from or transduced to annotations represented in a DAG. Recently, the potential to apply Linked Data formalisms to represent linguistic annotations, especially those residing on the web, has gained considerable interest, as this provides a uniform formalism for both query and data exchange. Again, the linked data model, serialized using RDF, is graph-based and therefore trivially mappable to other graph-based representations. The sections below provide examples of these approaches with attention to how they address the requirements for a generalized model outlined above.

5.1 XML formats for Standoff Annotations

GrAF and PAULA [50] provide examples of the standoff XML format. PAULA developed out of early drafts of the ISO TC37/SC4 Linguistic Annotation Framework (LAF) [77] and is hence closely related to GrAF. Both GrAF and PAULA are realized as standoff XML, which supports multi-layer corpora [50].

Both GrAF and PAULA serialize a graph-based model–i.e., a labeled directed acyclic (hyper)graph, in which the primary data structures are *nodes* and *edges*. In PAULA, various subtypes of these data structures are distinguished: a node is either a *token* (a character span in the primary data), a *markable* (a span of tokens), or a

[34] The nature of the referring pointer used may depend on the medium. For text, references to beginning and ending offsets ("virtual nodes" between characters) of a text span are standard.

struct (parent of other nodes). Edges are defined by the pair of nodes they connect: a *dominance relation* exists between a struct and its children; any other relation is classified as a *pointing relation*. The distinction between dominance and pointing relations enables development of convenient means to visualize and query the annotated data: for example, the appropriate visualization (hierarchical or relational) within a corpus management system can be chosen on the basis of the data structures alone, without requiring any external specifications. All types of nodes and edges can be labeled with one or more *features*, i.e., attribute-value pairs that express the actual annotations. In order to group nodes, edges, and labels, they are assigned a *namespace*.

The LAF/GrAF data model includes a similar, slightly simplified set of objects, visualized in Figure 9. PAULA's *terminals* correspond to GrAF's *regions*; otherwise, GrAF makes no distinction among nodes representing *markables* and *struct*s. Nodes are decorated with annotations, typically represented as simple feature structures (a group of one or more attribute-value pairs), but arbitrarily complex feature structures are also allowed. Nodes may have a *link* to a region or regions of primary data or an outgoing directed edge pointing to another node (annotation). In GrAF, edges signal a dominance relation between a node its children by default; child nodes are defined to be ordered constituents. Annotations on edges can specify a different interpretation, or, when an edge signals a relational ("pointing") annotation, it may specify the nature of that relation (e.g., anaphoric, alignment in parallel corpora, dependency). GrAF's *annotation spaces* perform the same function as PAULA's namespaces.

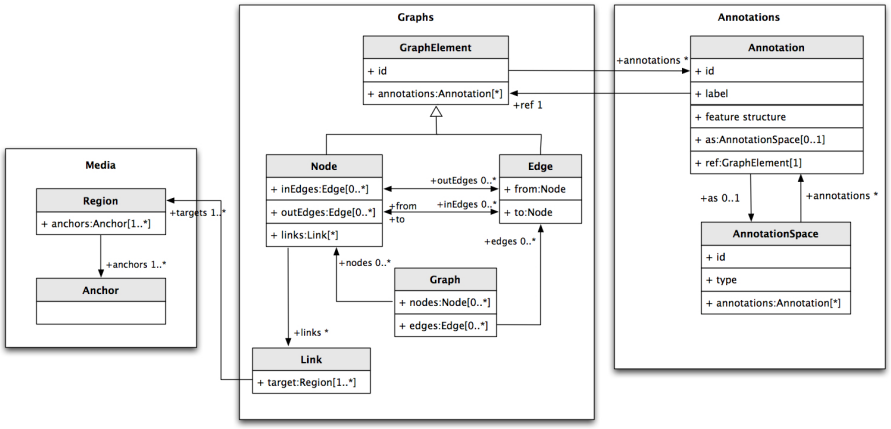

Fig. 9 UML representation of the LAF data model

The standoff XML approach is characterized by a separation between text and (different layers of) annotation. In LAF/GrAF, primary data is preserved in its original format, with no markup of any kind. PAULA/XML is not strictly stand-off but

rather a (weak) hybrid, as it allows minimal XML markup to be inserted into the primary data in order to use XLink/XPointer to references locations in the primary data. In both PAULA/XML and LAF/GrAF, the primary text is stored in a separate file, another file defines the minimal units that linguistic annotation can refer to, a third group of files comprises the actual annotations, and a fourth group of files contains associated metadata (optional in PAULA; obligatory in GrAF). GrAF also requires a *resource header* for a body of annotated data that specifies file name formats, dependencies among annotation files, namespaces for annotations of specific types or groupings/layers of files and annotations, and provides information about the processing software, segmentation rules, tag sets, etc.[35] The metadata requirement for data and annotations as well as the resource as a whole is intended to encourage principled and sufficient documentation that is lacking in many existing resources, with an eye toward enabling replicability of results, resource validation, and quality assessment.

A fragment of a PAULA file specifying the minimal units for reference from annotations is given below. This example contains XLink/XPointer references to a text file, but it may also include time-stamps or references to multi-modal content, or represent empty elements such as zero anaphors and traces:

```
<marklist xmnls=:xlink="http://www/w3.org/1999/xlink" type="tok"
      xml:base="tiger.syntax.procon.bae3umepro_040516.text.xml">
. . .
<mark id="tok_141"
   xlink:href="#xpointer(string-range(//body,'',809,5))"/>
<mark id="tok_142"
   xlink:href="#xpointer(string-range(//body,'',815,13))"/>
<mark id="tok_143"
   xlink:href="#xpointer(string-range(//body,'',829,4))"/>
<mark id="tok_144"
   xlink:href="#xpointer(string-range(//body,'',834,4))"/>
. . .
```

Table 1 PAULA specification of minimal units

GrAF requires definition of an *anchorType* in its resource header that specifies the format for anchors (pointers, references) into primary data and associates them with appropriate medium and file types. This can include character offsets or XLink/XPointers for text as well as anchors appropriate for image, audio, or video, and even XPath for documents including XML markup (although not recommended). An example is given in Figure 2.

The third type of files contain the actual annotations, typically, one per annotation type. Annotation content, i.e., labels and associated attribute/value pairs, may be given explicitly or (preferably) via the URI of an established repository or registry of linguistic categories (see Section 5.3). Annotations may be clustered according

[35] See [20] for more detailed information on the GrAF resource header.

```
<!-- Definitions in the resource header -->
<medium xml:id="text" type="text/plain" encoding="utf-8"
        extension="txt"/>
<medium xml:id="audio" type="audio" encoding="MP4"
        extension="mpg"/>
<medium xml:id="video" type="video" encoding="Cinepak"
        extension="mov"/>
<medium xml:id="video" type="image" encoding="jpeg"
        extension="jpg"/>
...
<anchorType xml:id="text-anchor" medium="text" default="true"
  lnk:href="http://www.xces.org/ns/GrAF/1.0/#character-anchor"/>
<anchorType xml:id="time-slot" medium="audio"
  lnk:href="http://www.xces.org/ns/GrAF/1.0/#audio-anchor"/>
<anchorType xml:id="video-anchor" medium="video"
  lnk:href="http://www.xces.org/ns/GrAF/1.0/#video-anchor"/>
<anchorType xml:id="image-point" medium="image"
  lnk:href="http://www.xces.org/ns/GrAF/1.0/#image-point"/>

<!-- Regions in the segmentation document -->
<region xml:id="r1" anchor_type="time-slot" anchors="980 983"/>
<region xml:id="r2" anchor_type="image-point"
        anchors="10,59 10,173 149,173 149,59"/>
<region xml:id="r3" anchor_type="video-hors="frame1(10,59)
        frame2(59,85) frame3(85,102)"/>
<region xml:id="r4" anchor_type="text-anchor"
        anchors="34 42"/>
```

Table 2 Region and anchor definitions in GrAF

to *layers* or *tiers* that represent a conceptual unit, e.g., all annotations generated from a particular source (such as TIGER/XML, MMAX, or ELAN) or annotations of a particular kind (such as syntax or coreference). In PAULA, layers may consist of group of files of different types identified by a shared id: if an annotation layer does not directly refer to a minimal unit file, one file can provide the elements of annotation (nodes, defined as either structs or markables), and another type of file can represent types of labels attached to these nodes. In GrAF, groups (which may be layers or tiers) of annotation types, files, individual annotations, ids, etc. are defined and named in the resource header.

Graph-based annotations that refer to the same primary data document can be easily merged, using standard graph merging algorithms followed by a validation step to guarantee the consistency of the resulting merged (hyper)graph[73]. Well-established algorithms for traversing and manipulating graphs can be applied to the merged graph to perform tasks such as common sub-tree analysis.

It should be noted that standoff annotations need not be represented in XML, although this is the most common means to represent standoff annotations intended for interoperable exchange due to its widespread use and the ready availability of XML processing tools. However, XML is extremely verbose and can increase the

size of annotated data by an order of magnitude, and standoff XML can be difficult for humans to read and manipulate. Other formats have been devised to get around these problems (e.g., the GrAF Compact Syntax[36], column-based formats.

5.2 Linked Data Representations

The evolution of technologies surrounding the Semantic Web has led to the possibility of representing linguistic data and annotations, as well as other linguistic resources such as lexicons, frame banks, and ontologies, as what is now termed *Linked Data*[37]. Linked Data exists on the web and, like much information on the web, is inter-connected to associated information (e.g., annotations) via URIs. Unlike general web hyper-links, Linked Data hyper-links are *typed*, thus providing a semantics for the relations the links represent. In the annotation scenario, this would allow for a link named "POS" from a token to an item in a list of categories, another named "lemma" to a lexicon entry, etc. Linked Data comes with a technological infrastructure that can be exploited by representing linguistic annotations in Linked-Data compliant formats such as the W3C Resource Description Framework (RDF)[38] and JSON/LD[39], which are themselves graph-based models. A major benefit of this approach is that off-the-shelf databases can be employed to store the data, and that a language for querying labeled directed graphs already exists (SPARQL 1.1[40]), and that the data can be exchanged in the same form as it is stored and processed.

From the perspective of computational linguistics, the Linked Data representation offers a number of advantages:

1. Using OWL/DL[41] reasoners, RDF data can be validated.
2. Using RDF as representation formalism, multi-layer corpora can be directly processed with off-the-shelf data bases and queried with standard query languages.
3. Information from different types of linguistic resources, e.g., corpora and lexical-semantic resources, can be combined using RDF. They can thus be queried with the same query language, e.g., SPARQL.
4. Linguistic corpora can be connected directly with repositories of reference terminology using RDF, thereby supporting the interoperability of corpora.

To address the need for a linked data framework for Natural Language Processing (NLP), the *NLP Interchange Format* (NIF) is an RDF/OWL-based format that

[36] http://graf.anc.org/gcs

[37] http://linkeddata.org

[38] http://www.w3.org/RDF/

[39] http://json-ld.org

[40] http://www.w3.org/TR/sparql11-query/

[41] http://www.w3.org/TR/owl-ref/

aims to achieve interoperability among NLP tools, language resources and annotations.[42] The NIF specification was released in an initial version 1.0 in November 2011[43]. The fundamental goal of NIF is to allow NLP tools to exchange annotations about text in RDF; therefore, the main prerequisite is that texts can be referenced with URIs in order to be used as *resources* (objects) in RDF statements. The NIF Core Ontology[44] provides classes and properties to describe the relations between substrings, text, documents and their URI schemes.

NIF addresses the annotation interoperability problem on three layers: the *structural* layer, the *conceptual* layer, and *access* layer. NIF is based on a Linked-Data-enabled URI scheme for identifying elements in (hyper-)texts that are described by the NIF Core Ontology (structural layer) and a selection of ontologies for describing common NLP terms and concepts (conceptual layer). NIF-aware applications produce output adhering to the NIF Core Ontology as REST services (access layer). As opposed to more centralized solutions such as *UIMA* [75] and *GATE* [74], NIF enables the creation of heterogeneous, distributed and loosely coupled NLP applications that use the Web as an integration platform. At the same time, annotated data conforming to NIF can be published as Linked Open Data as well, which opens possibilities for external reference, reuse, and further annotation.

Because RDF (and therefore NIF) is graph-based, it is virtually isomorphic to graph-based formats such as those described in the previous section. For example, a GrAF-to-RDF converter has been developed [49] and used to transduce the MASC corpus, a manually annotated sub-corpus of the Open American National Corpus (OANC) annotated for a wide range of linguistic phenomena [40] (see also Part II.I.c) to Linked Data form. Among others, MASC includes annotations for FrameNet frame elements and WordNet senses [43], as well as BabelNet senses [39]. In the GrAF version of MASC, WordNet senses are represented by sense keys as string literals; this representation can be trivially rendered as URI references pointing to an RDF version of WordNet. Similarly, FrameNet annotations can be linked to their descriptions in an OWL/DL version of FrameNet[45]. Such resources in Linked Data form would enable queries across the resources that were previously difficult or impossible. For example, it would be possible to search for sentences about *land*, i.e., "retrieve every sentence in MASC that contains a (WordNet-)synonym of *land*". Such queries can be used, for example, to develop semantics-sensitive querying engines for linguistic corpora.

Linked Data is only just coming of age, and its use as the primary representation format for linguistically-annotated data, especially where efficient and effective processing and searching is at issue, is likely inappropriate at least for the foreseeable future. However, given that RDF is a graph-based format, if the primary format for a resource conforms to the basic structural principles of generalized formats as outlined at the beginning of this section, adaptation to RDF/OWL will be trivial.

[42] For a more detailed description of NIF, see Chapter IV, Section 9 in this volume.

[43] http://nlp2rdf.org/nif-1-0/

[44] http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#

[45] The development of an OWL/DL version of FrameNet has been announced on the FrameNet site.

5.3 Repositories of Linguistic Concepts

A major benefit of inter-linkage among resources, either via RDF or simple hyperlinked references on the web, is the potential to move toward greater *semantic interoperability* [27] among linguistically-annotated resources. Formats such as those discussed in this chapter enable *syntactic interoperability* among resources, which relies on specified data formats to ensure that different systems can process exchanged information, but it provides no guarantee that the interpretation is the same. Semantic interoperability, on the other hand, enables different systems to interpret and process exchanged information in the same way–i.e., what is sent is exactly what is understood. Semantic interoperability is far harder to achieve for linguistically annotated data, not only because of the subtleties of the concepts used to describe linguistic phenomena, but also because of the variety of different theories and approaches that may come into the play.

Linked data resources provide a means to achieve greater semantic interoperability among linguistic annotations. A resource can be linked to a terminology or data category repository, and these community-defined data categories can be used to formulate queries that are independent of the annotation scheme using an abstract and well-defined vocabulary. In this way, linguistic annotations are not only syntactically interoperable (they use the same representation formalism), but also semantically interoperable (they use the same vocabulary).

Various repositories of linguistic terms have been established to serve as a reference point for linguistic annotations, so that terminology is unambiguously and consistently defined and common concepts are identified via mapping to terms in the repositories. A major effort in this area is ISOcat [59][46], a repository of linguistic categories maintained by ISO TC37/SC4. Terms in ISOcat are referenced by URI; an annotation can therefore use the URI reference for a linguistic label, feature, or attribute value rather than a simple string intended to represent a concept or category that has (in principle) been defined in some associated documentation. A related effort is the OLiA ontologies [47][47], which formalize numerous annotation schemes for morphosyntax, syntax and higher levels of linguistic description, and provide a linking to the morphosyntactic profile of ISOcat [48] with the General Ontology of Linguistic Description [54], and other terminology repositories. Although primarily concerned with the semantics of an annotation, the use of references to repositories of this kind has ramifications for the physical representation of the data: rather than a string representing a tag or label, the annotation includes a URI that points to terms defined and stored in a web-accessible location (cf. the requirements for NIF, stated above). An RDF interface has been proposed for ISOcat [64], which would encourage references to the repository from Linked Data representations of linguistically-annotated resources.

---

[46] See also Chapter IV, Section 6, in this volume.

[47] See also Chapter IV, Section 9, in this volume.

6 Choosing Representation Schemes

The choice of physical format for a linguistically-annotated resource should be dictated by the known and potential uses to which the resource may be put. The range of corpus types can be characterized as follows:

1. Corpora annotated in order to provide a general-purpose resource for use by others, with no specific application in mind, for example, the Penn and subsequent treebanks and discourse banks in other languages, the British, American, and other national corpora, etc.; and corpora developed in
2. Corpora designed with an eye toward both ease of development and ease of processing with different software, for example, the various corpora developed for the CoNLL and other shared task exercises.
3. Corpora developed primarily for in-house use or for access by others via a software interface, with no expectation of making them available for use by others (often for copyright reasons).

Any of the above types of corpora may contain multiple annotation types at different linguistic layers and even different modalities, and it may be expected that the developer or others will add annotations at a later stage (e.g., MASC); or they may be developed to provide annotations for a specific phenomenon (treebanks, discourse banks, time banks, etc.).

The representation choice for annotated corpora of type 1 is likely to be the most complex, especially if the corpus contains multiple annotations. A format able to accommodate the range of linguistic annotation types, provide a viable means to add, modify, and merge annotations, and maximally enable interoperability must necessarily make compromises between ease of use and expressivity in order to accommodate the widest range of annotation types and processing capabilities. Standoff XML formats, as described in Section 5, are sufficiently general to represent any linguistic annotation, and as such they serve well as a *pivot* for the interoperable exchange of data, by enabling trivial mappings into and out of other formats due to their grounding in a straightforward, graph-based underlying data model.

While the best choice of format for a general purpose corpus is likely to be standoff XML, corpora formatted this way are less well-suited to *working* with annotated data. Users typically rely either on in-house software with particular input/output requirements,or any of several available frameworks for processing annotated data (e.g., GATE, UIMA) that use their own internal formats. As mentioned in Section 2, transducers among widely-used formats such the the GATE and UIMA internal formats are increasingly available, thus making it possible to render a general-purpose corpus in standoff XML in the format required for well-known tools, and/or to move between tools as necessary. As for in-house formats, they can be mapped into and out of the pivot, more or less easily depending on the degree to which they conform to the graph-based model. Therefore, corpora of type 3 can be worked with using an in-house scheme or used in an annotation framework and, if necessary, transduced to the pivot for sharing or conversion to another scheme (using the pivot as the intermediary).

For a generalized corpus that is intended for access via the web, another option is a linked data representation, as described in Section refsec:background. Linked data representations employ existing and established standards with broad technical support (schemes, parsers, data bases, query language, editors/browsers, reasoners) and an active and comparably large community. For example, if datatypes are defined in OWL/DL, the validity of corpora can be automatically checked (according to the consistency constraints posited by an associated ontology such as POWLA), thus providing a possible solution to the semantic interoperability challenge for linguistic corpora [56].

Another common use of annotated corpora is to *store and query* the data. One means to do this is to store the data in a table representation and utilize relational databases for querying[53], A representative example of this approach is ANNIS; this tool provides a web browser-based search and visualization environment designed to access richly annotated corpora with heterogeneous annotation schemes [72, 83], which in its current implementation, ANNIS3, is based on a relational database (PostGreSQL).

Querying is also facilitated for corpora stored in a linked data format, which can be accessed using the SPARQL query language. Although relational data bases allow for flexible optimization and are thus well-suited to develop efficient corpus querying engines, they are based on fixed data base schemas. Accordingly, every modification of the data model requires a reinitialization of the data base, whereas an RDF database can updated without reinitialization. The RDF data model represents a superset of the data structures necessary to represent linguistic corpora, and therefore the relevant query operators exist.

As an important exception, transitivity has only recently been added to the SPARQL W3C recommendation (1.1, March 2013),[48] so that it is not widely supported yet. An alternative solution, however, is provided by OWL/DL-based inferences of transitive properties: If a property is defined as transitive, its transitive closure can be calculated using an OWL/DL reasoner, and the inferred triples can then be used in SPARQL queries.

In general, then, there is no "one size fits all" representation for linguistically annotated corpora, and the choice of format will be driven by both the immediate and foreseen needs of each project. It is common that a format is devised for in-house use that is easy to process and/or compatible with existing software. However, as it is increasingly likely that resources will be shared with others, it is worthwhile to make efforts, where possible, to ensure that an in-house format is amenable to transduction to generalized formats intended for interchange, for example, the graph-based models described above in Section 5. For existing formats, this means creating a mapping into and out of a format like LAF/GrAF, so that others may use transducers from that format to their chosen representation.

Creating new representation formats is less and less necessary these days, and it will become almost entirely unnecessary in the foreseeable future as more or less standardized tools and frameworks for creating processing linguistically-annotated

[48] http://www.w3.org/TR/2013/REC-sparql11-query-20130321/#propertypaths

resources come into widespread use. Should there be a motivation for creating a new format, however, several basic principles should be observed:

1. The format should be designed to reflect the abstract model underlying generalized graphs, which, as mentioned in Section 2, is the model used in not only pervasively in data structuring but also in database design, software design systems, and the semantic web.
2. All annotation information should be made *explicit*, that is, the burden of interpretation of given labels or structures should not be in the processing software.
3. An effort should be made to map labels and names to existing repositories (see Section 5.3).

References

1. DeRose, Steven J.: Grammatical Category Disambiguation by Statistical Optimization. Comput. Linguist. **14:1**, 31–39 (1988)
2. Church, Kenneth Ward: A stochastic parts program and noun phrase parser for unrestricted text. In: ANLC '88: Proceedings of the second conference on Applied natural language processing
3. Marcus, Mitchell P., Santorini, Beatrice, Marcinkiewicz, Mary Ann: Building a Large Annotated Corpus of English: The Penn Treebank. Comput. Linguist. **19:2**, 313–330 (1993)
4. Charniak, Eugene: A Maximum-entropy-inspired Parser. Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, 132–139, (2000)
5. Collins, Michael: Head-Driven Statistical Models for Natural Language Parsing. Comput. Linguist. **29:4**, 589–637, (2003)
6. ISO8879:1986: Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML). International Organization for Standardization, (1986)
7. Ide, Nancy, Bonhomme, Patrice, Romary, Laurent: XCES: An XML-based Standard for Linguistic Corpora. Proceedings of the Second International Language Resources and Evaluation Conference (LREC), 825–830 (2000)
8. Grishman, Ralph Sundheim, Beth: Message understanding conference - 6: A brief history. Proceedings of the International Conference on Computational Linguistics, 466–471 (1996)
9. Zampolli, Antonio: The PAROLE Project. In: R. Marcinkeviciene, N. Volz (eds.) The General Context of European Actions for Language Resources, Second European Seminar: Language Applications for Multilingual Europe, TELRE, 185–210 (1997)
10. Grishman, R. (ed.): Tipster Text Architecture Design. http://www-nlpir.nist.gov/related_projects/tipster/ (1998)
11. Bray, T., Paoli, J., Sperberg-McQueen, C.M. (eds.): Extensible Markup Language (XML) Version 1.0. W3C Recommendation. http://www.w3.org/TR/1998/REC-xml-19980210 (1998)
12. Ide, Nancy, Véronis, Jean: MULTEXT: Multilingual Text Tools and Corpora. In: Proceedings of the 15th International Conference on Computational Linguistics, COLING'94, 588–92 (1994)
13. Ide, Nancy, Véronis, Jean: What next after the Text Encoding Initiative? The need for text software. ACH Newsletter, Winter 1993, 1–12.
14. ISO 24612:2012: Language resource management – Linguistic Annotation Framework (LAF). International Organization for Standardization (2012)
15. Zipser, Florian, Romary, Laurent: A Model Oriented Approach to the Mapping of Annotation Formats using Standards. In: Proceedings of the Workshop on Language Resource and Language Technology Standards. 7–18 (2010)

16. Ide, Nancy, Romary, Laurent: A Registry of Standard Data Categories for Linguistic Annotation. Proceedings of the Fourth International Language Resources and Evaluation Conference (LREC'04), 135–138, (2004)
17. Kemps-Snijders, Marc, Windhouwer, Menzo, Wittenburg, Peter, Wright, Sue Ellen: ISOCat: Remodelling Metadata for Language Resources. In: International Journal of Metadata, Semantics and Ontologies, **4:4**, 261–276 2009
18. Cunningham, Hamish, Maynard, Diana, Bontcheva, Kalina, Tablan, Valentin : GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of ACL'02, (2002)
19. Ide, Nancy, Suderman, Keith: GrAF: A Graph-based Format for Linguistic Annotations. In: Proceedings of the Linguistic Annotation Workshop (LAW), 1–8 (2007)
20. Ide, Nancy, Suderman, Keith: The Linguistic Annotation Framework: A Standard for Annotation Interchange and Merging. *Language Resources and Evaluation* **48:3**, 395-418 (2014)
21. Bird, Steven, Liberman, Mark: A formal framework for linguistic annotation. Speech Communication **33:1-2**, 23–60 (2001)
22. Ide, Nancy, Suderman, Keith, Simms, Brian: ANC2Go: A Web Application for Customized Corpus Creation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC), (2010)
23. Ide, Nancy: Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In: Proceedings of the First International Language Resources and Evaluation Conference (LREC), 463–70 (1998)
24. Ferrucci, David, Lally, Adam: UIMA: An architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering **10:3-4**, 327–348 (2004)
25. Neumann, Arne, Ide, Nancy, Stede, Manfred: Importing MASC into the ANNIS linguistic database: A case study of mapping GrAF. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse (LAWIV-ID), 98–102 (2013)
26. Chiarcos, Christian, Hellmann, Sebastian, Nordhoff, Sebastian : Towards a linguistic linked open data cloud: The Open Linguistics Working Group. TAL, 245–275 (2011)
27. Ide, Nancy, Pustejovsky, James: What Does Interoperability Mean, Anyway? Toward an Operational Definition of Interoperability for Language Technology. Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010), (2010)
28. Stede, Manfred, Neumann, Arne: Potsdam Commentary Corpus 2.0: Annotation for Discourse Research. Proceedings of the International Language Resources and Evaluation Conference (LREC), (2014)
29. Dipper, Stefainie: XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In: Proceedings of Berliner XML Tage 2005, 39–50. Berlin (2005)
30. Diewald, Nils, Sthrenberg, Maik, Garbar, Anna, Goecke, Daniela: Serengeti - Webbasierte Annotation semantischer Relationen. Journal for Language Technology and Computational Linguistics 23(2), 74-93 (2008)
31. Teufel, Simone, Moens, Marc: Summarizing Scientific Articles – Experiments with Relevance and Rhetorical Status. Computational Linguistics 28(4), 409–445 (2002)
32. Müller, Christoph, Strube, Michael: Multi-Level Annotation of Linguistic Data with MMAX2. In: Sabine Braun, Kurt Kohn, Joybrato Mukherjee (Eds.): Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods. Frankfurt: Peter Lang, 197–214 (2006)
33. Mann, William, Thompson, Sanda: Rhetorical structure theory: Towards a functional theory of text organization. TEXT 8, 243-281 (1988)
34. Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: Linguistic interpretation of a German corpus. Research on Language and Computation 2 (4), 597-620 (2004)
35. Brants, Thorsten, Skut, Wojciech, Krenn, Brigitte: Tagging Grammatical Functions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97). Providence, RI (1997)

36. Baker, Collin F., Fillmore, Charles J., Lowe, John B.: The Berkeley FrameNet project. COLING-ACL '98: Proceedings of the Conference. Montreal, Canada, 86-90 (1998)
37. Banski, Piotr, Przepiórkowski, Adam: Stand-off TEI Annotation: The Case of the National Corpus of Polish. In Proceedings of the Third Linguistic Annotation Workshop (LAW III). Suntec, Singapore, 64-67 (2009)
38. Chiarcos, Christian, Ritz, Julia, Stede, Manfred: By all these lovely tokens... Merging conflicting tokenizations. Language Resources and Evaluation 46(1), 53–74 (2012)
39. Moro, Andrea, Navigli, Roberto, Tucci, Francesco Maria, Passonneau, Rebecca J.: Annotating the MASC Corpus with BabelNet. In In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC). (2014)
40. Ide, Nancy, Baker, Collin, Fellbaum, Christiane, Passonneau, Rebecca: The Manually Annotated Sub-Corpus: A Community Resource for and by the People. In Proceedings of the ACL 2010 Conference Short Papers. Uppsala, Sweden, 68–73 (2010)
41. Baker, Collin F., Fellbaum, Christiane: WordNet and FrameNet as Complementary Resources for Annotation. In Proceedings of the Third Linguistic Annotation Workshop(LAW III), Suntec, Singapore, 125–129 (2009)
42. Chiarcos, C.: A generic formalism to represent linguistic corpora in RDF and OWL/DL. In Proceedings of Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC). Istanbul, Turkey (2012)
43. Baker C, Fellbaum C (2009) WordNet and FrameNet as complementary resources for annotation. In: Third Linguistic Annotation Workshop (LAW-2009), Suntec, Singapore, pp 125–129
44. Bird S, Liberman M (2001) A formal framework for linguistic annotation. Speech Communication 33(1-2):23–60
45. Bow C, Hughes B, Bird S (2003) Towards a general model of interlinear text. In: Proceedings of EMELD workshop, pp 11–13
46. Carletta J, Evert S, Heid U, Kilgour J (2005) The NITE XML Toolkit: data model and query. Language Resources and Evaluation Journal (LREJ) 39(4):313–334
47. Chiarcos C (2008) An ontology of linguistic annotations. LDV Forum 23(1):1–16
48. Chiarcos C (2010) Grounding an ontology of linguistic annotations in the Data Category Registry. In: Workshop on Language Resource and Language Technology Standards (LR&LTS 2010), held in conjunction with LREC 2010, Valetta, Malta
49. Chiarcos C (accepted) A generic formalism to represent linguistic corpora in RDF and OWL/DL. In: 8th International Conference on Language Resources and Evaluation (LREC-2012)
50. Chiarcos C, Dipper S, Götze M, Leser U, Lüdeling A, Ritz J, Stede M (2008a) A Flexible Framework for Integrating Annotations from Different Tools and Tagsets. TAL (Traitement automatique des langues) 49(2):217–246
51. Chiarcos C, Dipper S, Gtze M, Leser U, Ldeling A, Ritz J, Stede M (2008b) A flexible framework for integrating annotations from different tools and tagsets. TAL (Traitement automatique des langues) 49
52. Chiarcos C, Ritz J, Stede M (accepted) By all these lovely tokens ... merging conflicting tokenizations. Journal of Language Resources and Evaluation (LREJ)
53. Eckart K, Riester A, Schweitzer K (2012) A discourse information radio news database for linguistic analysis. In: Chiarcos C, Nordhoff S, Hellmann S (eds) Linked Data in Linguistics, Springer
54. Farrar S, Langendoen DT (2010) An OWL-DL implementation of GOLD: An ontology for the Semantic Web. In: Witt A, Metzing D (eds) Linguistic Modeling of Information and Markup Languages, Springer, Dordrecht
55. Goodwin C, Heritage J (1990) Conversation analysis. Annual review of anthropology pp 283–307
56. Ide N, Pustejovsky J (2010) What does interoperability mean, anyway? Toward an operational definition of interoperability. In: Second International Conference on Global Interoperability for Language Resources (ICGL 2010), Hong Kong, China
57. Ide N, Suderman K (2007) GrAF: A graph-based format for linguistic annotations. In: Proceedings of The Linguistic Annotation Workshop (LAW) 2007, Prague, pp 1–8

58. Ide N, Romary L, de la Clergerie E (2003) International Standard for a Linguistic Annotation Framework. In: Proceedings of HLT-NAACL'03 Workshop on the Software Engineering and Architecture of Language Technology, Edmonton, Canada, pp 25–30
59. Kemps-Snijders M, Windhouwer M, Wittenburg P, Wright S (2009) ISOcat: Remodelling metadata for language resources. International Journal of Metadata, Semantics and Ontologies 4(4):261–276
60. Müller C, Strube M (2006) Multi-level annotation of linguistic data with mmax2. In: Braun S, Kohn K, Mukherjee J (eds) Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods, Peter Lang, Frankfurt, Germany, pp 197–214
61. Schmidt T (2003) Visualising linguistic annotation as interlinear text. Sonderforschungsbereich 538
62. Schmidt T, Duncan S, Ehmer O, Hoyt J, Kipp M, Loehr D, Magnusson M, Rose T, Sloetjes H (2009) An exchange format for multimodal annotations. In: Multimodal corpora, Springer, pp 207–221
63. Wightman C, Price P, Pierrehumbert J, Hirschberg J (1992) Tobi: A standard for labeling english prosody. In: Proceedings of the 1992 International Conference on Spoken Language Processing, ICSLP, pp 12–16
64. Windhouwer M, Wright S (2012) Linking to linguistic data categories in ISOcat. In: Chiarcos C, Nordhoff S, Hellmann S (eds) Linked Data in Linguistics, Springer, Heidelberg, pp 99–107
65. Wittenburg P, Lenkiewicz P, Auer E, Lenkiewicz A, Gebre BG, Drude S (2012) Av processing in ehumanities–a paradigm shift. In: Digital Humanities 2012 Conference, vol 2
66. Zeldes A, Ritz J, Ldeling A, Chiarcos C (2009) ANNIS: A search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009, Liverpool, UK
67. Bradshaw, Julie, Burridge, Kate and Clyne, Michael (2010) 'The Monash Corpus of Spoken Australian English.' Proceedings of the 2008 Conference of the Australian Linguistics Society., 2123/7099
68. Schmidt, T.; Elenius, K. and Trilsbeek, P. (2010) Multimedia Corpora (Media encoding and annotation). Draft submitted to CLARIN WG 5.7. as input to CLARIN deliverable D5.C-3 "Interoperability and Standards"
69. Bird, S., Liberman, M.: A formal framework for linguistic annotation. Speech Communication **33**(1-2), 23–60 (2001)
70. Bow, C., Hughes, B., Bird, S.: Towards a general model of interlinear text. In: Proceedings of EMELD workshop, pp. 11–13 (2003)
71. Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML Toolkit: data model and query. Language Resources and Evaluation Journal (LREJ) **39**(4), 313–334 (2005)
72. Chiarcos, C., Dipper, S., Gtze, M., Leser, U., Ldeling, A., Ritz, J., Stede, M.: A flexible framework for integrating annotations from different tools and tagsets. TAL (Traitement automatique des langues) **49** (2008)
73. Chiarcos, C., Ritz, J., Stede, M.: By all these lovely tokens ... merging conflicting tokenizations. Journal of Language Resources and Evaluation (LREJ) (accepted)
74. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: ACL (2002). DOI 10.3115/1073083.1073112. URL http://www.aclweb.org/anthology/P02-1022
75. Ferrucci, D., Lally, A.: UIMA: An architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering **10**(3/4), 327–348 (2004)
76. Goodwin, C., Heritage, J.: Conversation analysis. Annual review of anthropology pp. 283–307 (1990)
77. Ide, N., Romary, L., de la Clergerie, E.: International Standard for a Linguistic Annotation Framework. In: Proceedings of HLT-NAACL'03 Workshop on the Software Engineering and Architecture of Language Technology, pp. 25–30. Edmonton, Canada (2003)
78. Ide, N., Suderman, K.: GrAF: A graph-based format for linguistic annotations. In: Proceedings of The Linguistic Annotation Workshop (LAW) 2007, pp. 1–8. Prague (2007)
79. Schmidt, T.: Visualising linguistic annotation as interlinear text. Sonderforschungsbereich 538 (2003)

80. Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., Sloetjes, H.: An exchange format for multimodal annotations. In: Multimodal corpora, pp. 207–221. Springer (2009)
81. Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J.: Tobi: A standard for labeling english prosody. In: Proceedings of the 1992 International Conference on Spoken Language Processing, ICSLP, pp. 12–16 (1992)
82. Wittenburg, P., Lenkiewicz, P., Auer, E., Lenkiewicz, A., Gebre, B.G., Drude, S.: Av processing in ehumanities–a paradigm shift. In: Digital Humanities 2012 Conference, vol. 2 (2012)
83. Zeldes, A., Ritz, J., Ldeling, A., Chiarcos, C.: ANNIS: A search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009. Liverpool, UK (2009)