Representing and querying multi-layer annotated corpora

Elke Teich
Applied Linguistics
University of the Saarland
Saarbrücken, Germany
⟨e.teich@mx.uni-saarland.de⟩

Silvia Hansen
Computational Linguistics
University of the Saarland
Saarbrücken, Germany
⟨hansen@coli.uni-sb.de⟩

Peter Fankhauser
FhG-IPSI
Darmstadt
Germany
⟨fankhauser@ipsi.fhg.de⟩

Abstract

The goal of the present paper is to review the methods employed in multi-layer corpus representation and querying and to identify the more fundamental issues involved in the task. Asking questions such as: What are the linguistic requirements on multi-layer corpora? Do the data structures commonly employed mirror the needs of linguistic representation appropriately? Should different layers of annotation be kept separate, or should and can they be merged? How and at which granularity can we align/integrate multiple layers of annotation? If expressively more powerful data structures than trees/forests are required, then what kinds of mechanisms are needed to query such data structures?, we propose an XML-based approach to the representation and querying of multi-layer corpora.

1 Introduction

There has been an increasing activity recently in building corpora annotated with multiple layers of linguistic information, i.e., text and speech corpora annotated for different kinds of linguistic units (syllables, words, clauses, utterances etc.) at different linguistic strata including phonology, grammar and semantics.

As corpora become more comprehensive in terms of the kinds of linguistic information they make available, they potentially become an increasingly interesting resource for linguists more generally. However, their accessibility remains limited: typically, only selected linguistic perspectives are focused upon, proprietary formats are employed, and only proprietary tools are supported for search and analysis. It is only recently that techniques for the representation of *multiple* perspectives on corpora are aimed at (cf. e.g., (Bird and Harrington, 2001)), such that multi-layer corpora can be maintained, queried and analyzed in an integrated fashion. It seems to us that now is a good point in time for a critical appraisal of previous and on-going work, in order to identify the more fundamental issues involved in multi-layer corpus representation and querying.

We proceed in the following way. Section 2 presents our own starting situation. We focus on two cases of linguistic analysis, one from multilingual analysis and one from spoken language analysis. We describe how we employed different kinds of tools for different kinds of annotation, each basically using a different kind of format and data structure. We show how we created a uniform and integrated representation for the different kinds of annotation outputs on the basis of XML (W3C, 2000) for the first analysis case. However, for the second kind of analysis problem, an integrated representation turns out unfeasible. We conclude Section 2 with a set of requirements on multi-layer corpus representation derived from our analysis needs.

Section 3 takes this set of requirements as a starting point and discusses each of them in relation to the two methods of multi-layer corpus representation that are predominant at the moment. One is committed to the concept of dominance and uses directed acyclic graphs (as employed in 'tree banks' and in general-purpose linguistic database systems such as e.g., EMU (Cassidy and Harrington, 2001)); the other is committed to the concept of temporality—this is the annotation graph as proposed by (Bird and Liberman, 2001). In the remainder of Section 3 we outline an approach to the representation, maintenance and querying of multi-layer corpora that is guided by our requirements and builds upon readily available tools for XML as much as possible.

Section 4 concludes the paper with a short summary and some directions for future research.

2 Starting point and requirements: multiple perspectives on a corpus

We have encountered the need for multi-layer corpora in two contexts of linguistic analysis, one being the contrastive-linguistic analysis of translations and otherwise multilingually comparable texts (Teich and Hansen, 2001; Hansen and Teich, 2001), the other being the analysis of the grammar and prosody of spoken language in the context of speech generation (Teich et al., 2000). In the first kind of application, what we are mainly interested in are the specific properties of translations compared to original texts in the same language as the target language (cf. (Baker, 1995; Steiner, 2000; Teich, 2001)): If translations have specific properties, what are they and how do they come about? One hypothesis that has been formulated is that in translations, the source language is bound to "shine through". For instance, comparing English and German, English Subjects are more versatile in the kinds of semantic properties they can take on, nonagentive Subjects being quite common in English whereas in German they are not. Consider examples (1)-(3). A direct translation of a nonagentive clause as in (1) into the German (2) comes across as rather metaphorical. To compensate, a German translation may resort to a transposition as exemplified in (3), but more direct translations such as (2) do occur as well (cf. (Teich, 2001)).[1]

(1) *Textbooks write this process as HCl H+ + Cl-.*

(2) *Lehrbücher beschreiben diesen Prozess als HCl H+ + Cl-.*
textbooks describe this process as HCl H+ + Cl-

(3) *In Lehrbüchern wird dieser Prozess als HCl H+ + Cl- dargestellt.*
in textbooks is this process as HCl H+ + Cl- presented

In this example, we are looking at three kinds of linguistic information: a semantic feature of the clause (nonagentive), a functional-grammatical feature attributed to the nominal phrase (subject), and a surface-syntactic feature of the word (part-of-speech: nominal). We thus need to annotate the corpus for these kinds of information in order to be able to extract instances of this kind of phenomenon.

Doing so, we proceeded in the following way.

For part-of-speech (pos) tagging, we used TnT (Brants, 2000), a statistical tagger. TnT can use different kinds of tag sets and has been trained on a number of languages, including English and German. The tag set used for English is based on the Susanne tag set (Sampson, 1995); the one for German is the Stuttgart-Tübingen tag set (Hinrichs et al., 1995). The result of the tagging process is represented in a TSV format. For example (1) the pos-annotation looks as follows:

```
Textbooks       NN2
write           VVO
this            DD1
process         NN1
as              II
HCl H+ + Cl-    NP1
.               YC
```

Syntactic annotation has been carried out manually, but potentially, an automatic parser can be employed for a shallow phrase structure analysis. One such system for German which includes shallow parsing and integrates the TnT tagger is the ANNOTATE system (Plaehn and Brants, 2000). Syntactic analysis in ANNOTATE in addition covers grammatical functions. The output is represented as a tree, where terminal nodes carry pos labels, non-terminal nodes carry phrase structure labels and edges carry grammatical function labels. The annotation of example (1) looks as follows:[2]

```
(S
  (NP-SB
      (NN2-NK Textbooks)
  )
  (VP-HD
      (VVO-HD write)
    (NP-OA
      (DD1-NK this)
      (NN1-NK process)
    )
    (PP-MO
      (II-AC as)
      (NP1-NK HCl H++ Cl-)
    )
  )
  (YC .)
)
```

For the annotation of semantic linguistic features, such as agentivity, tools are needed which

[1] The example is from a parallel corpus of English original texts from *Scientific American* and their translations into German from *Spektrum der Wissenschaft.*

[2] This is the same as the Penn Treebank format (Marcus et al., 1993).

```
<?xml version="1.0">
  <clause semfeat="nonagentive">
     <phrase synform="NP" synfunc="SB">
        <word pos="NN2">Text books</word>
     <phrase synform="VP" synfunc="HD">
        <word pos="VV0">write</word>
        <phrase synform="NP" synfunc="OA">
           <word pos="DD1">this</word>
           <word pos="NN1">process</word>
        </phrase>
        <phrase synform="PP" synfunc="MO">
           <word pos="II">as</word>
           <word pos="NP1">HCl H+ + Cl-</word>
        </phrase>
     </phrase>
     <word pos="YC">.</word>
  </clause>
```

Figure 1: Integrated XML representation

support manual annotation. One such tool is Coder (O'Donnell, 1995). Coder provides the possibility of defining annotation schemes and guides the user through the annotation process. Text annotated with Coder is represented in an XML-like format. For example (1), this looks as follows:

```
<segment features=''clause nonagentive''> Text books write this
process as HCl H+ + Cl-. </segment>
```

If we want to arrive at an integrated representation of these different kinds of annotation, so as to make the corpus searchable with reference to more than one kind of annotation at the same time, the different output formats have to be merged into one uniform representation. For this purpose, we employed XML: We defined a document-type definition (DTD) that covers the different kinds of annotation, and performed a number of format transformations into XML. The integrated XML-encoded version of our example looks as shown in Figure 1.

Using XML, we have achieved both a uniform and an integrated representation of three different kinds of annotation, and we have made sure that instances of annotated text are valid against a DTD. Also, the annotated corpus is searchable in the way we would like it to be, i.e., in querying it we can make reference to more than one kind of annotation at a time, provided a corresponding query facility is available (e.g., the MATE system's (Mengel and Lezius, 2000) query meachanism Q4M (Heid and Mengel, 1999)).

While for this kind of analysis problem an integrated representation is a possible solution, there are other kinds of linguistic analysis questions where an integrated representation becomes problematic or even unfeasible. From a linguistic point of view, this situation arises when the underlying relation we want to model is not a hierarchical one. In the example we have discussed, we have taken the constituency relation as the basis and represented the other kinds of information as attributes of some basic phrase structure categories (e.g., ⟨`clause semfeat=''nonagentive''` ⟨/`clause`⟩), a common practice in many grammar models. The 'layers' are clause, phrase, word, where phrase is regarded as a constituent of clause and word is regarded as a constituent of phrase. Linguistically, these 'layers' are at the same linguistic stratum, namely the stratum of grammar. We can see in this example that it is possible to model the relation between 'layers' in the sense of phrase structure levels in this way. However, there are clearly many other kinds of linguistic relations which do not involve simple hierarchical relations such as this one. One case in point is the relation between syntactic units and intonation units. Here, we look across two linguistic *strata*: grammar and phonology. This means that the units we are interested in are not in a constituency or otherwise hierarchical relation; but, according to most linguistic models, this relation is much more complex—so complex that it needs to be modeled by some kind of rule apparatus.[3] More concretely, in the case of syntactic units and intonation units, when we try to relate one to another, we cannot even be sure that they are co-extensive, i.e., share the same boundaries.

Suppose we are interested in whether there is a correlation between clause complexity (i.e., whether a clause is continuing or not, and if so, whether it stands in a paratactic or a hypotactic relation to the previous clause) and the selection of tone (intonation contour). Consider the example below, segmented into clauses and annotated for clause complexity in (4) and segmented into intonation units and annotated for

[3] There is a considerable amount of literature on the syntax-phonology interface from various kinds of theoretical perspectives, ranging from Government & Binding (Selkirk, 1984) over Categorial Grammar (Steedman, 2000) to Systemic Functional Grammar (Halliday, 1967). This simply shows that the relation we deal with here is not a trivial one, and therefore, we cannot expect to have a simple solution for annotation either.

intonation contours in (5).[4]

(4) (`clause:cont-init` *okay well you turn right*) (`clause:cont+para` *and you go along the corridor*) (`clause:cont+para` *and you turn left into the second little corridor*) (`clause:cont+hypo` *and as soon as you do that*) (`clause:fin` *the office will be straight on your right*)

(5) *//*`tone:t3` *okay well you turn RIGHT//*`tone:t3` *and you go along the CORidor and//*`tone:t3` *you turn LEFT//*`tone:t3` *into the second little CORridor//*`tone:t2` *and as soon as you do THAT//*`tone:t1` *the office will be straight on your RIGHT//*

This example illustrates the phenomenon of *overlapping segments*. In the syntactic segmentation of the chunk *and you go along the corridor and you turn left* we have: (*and you go along the corridor*) (*and you turn left*), whereas in the phonological segmentation we have: (*and you go along the corridor and*) (*you turn left*). Also, each of the two kinds of segments has a set of attributes that is disjoint from the set that holds for the other.

If we were to integrate the two kinds of annotation into a representation as shown in Figure 1, we would run into problems, because there is no dominance relation involved between the two kinds of segments, but rather one of inter-stratal realization. A more adequate representation is to leave the two 'layers' of annotation separate and align them at a particular level (e.g., at the level of words, syllables or even characters). A representation with separate layers is possible using a technique that has been termed *stand-off markup* in XML-based approaches (Thompson and McKelvie, 1997). For the example above this is shown in Figure 2 in Section 3 below.

The more general requirements that emanate from our concrete experiences are the following. First, the underlying data model should be **expressive** enough to capture the major kinds of relationships relevant for our needs of linguistic analysis. These comprise dominance and stratificational relationships (as exemplified by the two analysis cases above), but also referent-antecedent relationships, or aligned parallel text and aligned text-sound. Second, notwithstanding expressive power, the data model should be as **simple** and orthogonal as possible to facilitate (also manual) annotation, maintenance, sharing, and analysis. Third, since in our application context we typically employ quite heterogeneous annotation tools (basically all of them use different formats and data structures), it is doubly important to be able to check the **integrity** of an integrated/aligned representation. We believe that our situation will, and should in fact, become more and more commonplace, i.e., linguists will increasingly employ existing tools for corpus annotation rather than implementing their own, and the issue of the integrity of a corpus resource in which different kinds of annotations have been produced by different kinds of programs will therefore need serious attention. Finally, to enable effective corpus analysis we need expressive and scalable **search** facilities that take into account the linguistically relevant relationships and in particular allow us to express queries against multiple perspectives imposed on a corpus.

The following section reviews the two currently predominant methods of multi-layer corpus representation, discussing how they meet these requirements. On that basis, we outline an XML-based approach to multi-layer corpus representation and querying.

3 Multi-layer corpora: representation and querying

3.1 State-of-the-art

There are currently basically two kinds of approaches in use for representing and querying multi-layer corpora: Annotation graphs and (ordered) directed acyclic graphs.

Annotation graphs (AGs) (Bird and Liberman, 2001) emphasize the temporal dimension

[4] The example is taken from a corpus of induced natural speech in the domain of route descriptions compiled by Sandra Williams at MRI, Macquarie University, Sydney.—This is an ad-hoc representation which simply serves for illustration. The segment boundaries in (4) are denoted by brackets and the labels code the following categories: `cont-init` = continuing clause in sentence-initial position, `cont+para` = continuing clause parataxis, `cont+hypo` = continuing clause hypotaxis, `fin` = clause in sentence-final position. The segment boundaries in (5) are denoted by // and the labels are tone labels as proposed in (Halliday, 1967). Also, the syllable carrying the nuclear pitch accent is indicated by capital letters.

of (speech-) corpora. A corpus is overlayed with one or more timelines. Each timeline is divided up into nodes at certain time offsets. Nodes are connected by directed arcs, which segment the corpus at various levels of granularity. Every arc can be adorned with a variable length tuple of attribute-value pairs to represent its type, its transcribed content, and possibly other metainformation. AGs are closed under union, intersection and difference, they can be naturally represented as relational tables, and a fairly expressive query language can be defined on the basis of datalog (Bird et al., 2000).

Annotation graphs represent the temporal order of segments very naturally, and multiple perspectives on a corpus with possibly intersecting segmentations are well coordinated via reference to a common timeline. However, hierarchical structure is only represented indirectly: An arc a_1 from node n_{11} to node n_{12} contains an arc a_2 from n_{21} to n_{22} if $time(n_{11}) \leq time(n_{21})$ and $time(n_{12}) \geq time(n_{22})$. If $time(n_{11}) = time(n_{21})$ and $time(n_{12}) = time(n_{22})$, this becomes ambiguous; a_1 contains a_2 and a_2 contains a_1. Furthermore, to check whether a_1 contains a_2 as a direct child requires negation: a_1 contains a_2 directly if a_1 contains a_2 and there exists no a_3 such that a_1 contains a_3 and a_3 contains a_2. This is not a principle problem of annotation graphs, though. Hierarchical structure can be represented without information loss by introducing special child and parent arcs. For a particular hierarchical perspective the arcs then essentially represent the trace of a push-down tree-automaton (Neumann, 1999), which is a convenient formalism to describe the semantics of XML-DTDs and XML-Schemas.

Trees, and in particular XML as their popular incarnation, treat sequential (temporal) order and hierarchical structure on equal grounds. However, trees fail to represent multiple intersecting hierarchies. Several projects (MATE (McKelvie et al., 2001), EMU (Cassidy and Harrington, 2001), GODDAG (Sperberg-McQueen and Huitfeldt, 2001)) have tried to overcome this limitation.

MATE employs two main techniques for representing intersecting hierarchies in XML: XPointers and time offsets. With XPointers, one hierarchy is represented as the primary hierarchy in XML, and each element is equipped with a unique id. Other hierarchies can be superimposed on elements of the primary hierarchy by referencing to (ranges of) their unique ids. In the internal representation and in Q4M, the MATE query language, intersecting hierarchies are represented as an **ordered directed acyclic graph** (ODAG). Thereby, the inherent asymmetry between primary and secondary hierarchies is overcome. Time offsets are represented by means of start- and end-attributes, and can be queried with a rich set of operators. Also GODDAG and TEXMECS (Sperberg-McQueen and Huitfeldt, 2001) are based on ODAGs, and in addition investigate techniques to serialize intersecting and possibly interrupted hierarchies superimposed on a corpus and to generalize validation for (XML-)trees to validation for ODAGs.

How well then do these two approaches fulfil our requirements?

Expressiveness. AGs, augmented with arcs for representing hierarchical structure, and ODAGs, augmented with time-offsets, are essentially equivalent, i.e., one can map between them without information loss. Arcs in AGs correspond to nodes in ODAGs, child/parent-arcs in AGs correspond to child-edges in ODAGs (or more precisely, to the first-child and last-parent edge), the time-offsets of the start- and end-nodes of an arc in AGs correspond to start- and end-attributes of a node in ODAGs. Both representations can support general (hyper-)graphs, needed to represent, for instance, syntactic agreement, referent-antecedent relationships, or parallel aligned text by means of shared attribute-values.

Simplicity. Compared with ODAGs, AGs generally appear simpler and more orthogonal. This is mainly for two reasons. Firstly, pure AGs deliberately do not distinguish hierarchical structure from temporal containment. This results in one design choice less to make, and it encourages modularity of annotations, e.g., it avoids spurious hierarchical relationships across layers to represent just temporal containment. Secondly, ODAGs are often discussed as some variation of XML, which gives rise to even more design choices: By which concrete syntactic means shall intersecting hierarchies be represented? Which aspect of a node shall be chosen as an element-name, and which aspects shall be represented by attributes? At the level of graphs, these choices vanish, and AGs and ODAGs are of comparable orthogonality.

Integrity. Manual annotation of corpora, but also annotation by means of poorly documented tools and transformation scripts is errorprone. For ODAGs there exists a readily available toolset to detect typical errors by means of constraints in the form of DTDs, and various generalizations such as XML-Schema or Relax-NG. The constraint mechanisms for AGs, and their underlying relational representation—essentially key-, value-, and (referential) integrity constraints—are less well explored for the purpose of detecting typical annotation errors. A combination of the grammar and rule based validation mechanisms for XML and the constraint mechanisms for relational tables to check integrity of multiple intersecting hierarchies is worthwhile to explore.

Searchability. Augmented AGs can represent both sequential order and hierarchical structure of annotations by means of relational tables, without resorting to auxiliary positional attributes to represent sequential order. In contrast, generic relational mappings for XML (Florescu and Kossman, 1999) and more generally ODAGs need to represent sequential order with a positional attribute. Therefore, the well established mechanisms for querying relational tables are more readily applicable to AGs than to ODAGs, even though querying AGs requires rather heavy transitive closures. The real strength of AGs relational representation is updatability, because insertion or deletion of arcs does not require costly recomputation of positional attributes or shifting of nodes. The special purpose query language devised for MATE's ODAGs is also sufficiently expressive, but unfortunately not backed up by an even moderately scalable implementation. In Section 3.3 we will explore how XQuery, a query language for XML, currently standardized by the W3C, is applicable to querying ODAGs represented as XML.

3.2 Representation

Our approach to corpus representation draws on both AGs and ODAGs. From AGs we adopt the principle of modularity, i.e., we keep independent layers of annotation separate, even if they can in principle be merged into a single hierarchy. From ODAGs we adopt the possibility to explicitly represent structural hierarchies—if they are meaningful. Like in MATE, here we have chosen XML as the concrete data format to maintain and represent annotations, in order to take advantage of the rich set of readily available tools to edit, validate, transform, and query XML.

In more detail, our approach is as follows:

(1) The corpus is represented as a flat text file.

(2) Each layer is represented as a separate XML file marking up the corpus[5]. To control the errorprone process of manual and semi-automatic annotation, we deploy DTDs at various degrees of specificity.

(3) Each element containing a text-node is equipped with two attributes `start` and `end`, which represent the character offset of its content in the (whitespace-normalized) corpus. Elements higher up in the hierarchy "inherit" `start`- and `end`-attributes from the first and last text-node descendants, respectively. The corpus thus serves as a normative basis for aligning several layers.[6]

(4) To share elements among multiple hierarchies, the elements can be optionally equipped with a unique id.[7]

(5) More complex relationships needed to represent e.g., syntactic agreement, referent-antecedent relations, or parallel aligned text are also represented via shared attribute-values (c.f. AGs' "equivalence classes")[8].

Figure 2 gives an example of the English portion of the parallel corpus fragment introduced in Section 2. The two annotation layers—part-of-speech and phrase structure—are kept separate, and (automatically) aligned with the base text. We deliberately keep the redundant text-content in the aligned versions to facilitate human inspection. With the text-content discarded, the representation essentially amounts to a variation of stand-off markup (Thompson

[5] Each layer also uses a separate namespace, which is not given in the examples to keep them simple to read.

[6] We produce these references by means of a simple (generic) XSLT-stylesheet, but they could also be produced by a clever editor, or by tools used for automatic annotation.

[7] This can also be accomplished by means of an XSLT-stylesheet.

[8] To keep a low technology profile, we refrain from using XLink or XPointer.

```
<?xml version="1.0"?>
<text>
Text books write this process as HCl H+ + Cl-.
<text>

<?xml version="1.0"?>
<words>
  <word pos="NN2" nc="textbooks" start="1" end="10">Textbooks</word>
  <word pos="VV0" start="12" end="16">write</word>
  <word pos="DD1" start="18" end="21">this</word>
  <word pos="NN1" start="23" end="29">process</word>
  <word pos="II" start="31" end="32">as</word>
  <word pos="NP1" start="34" end="45">HCl H+ + Cl-</word>
  <word pos="YC" start="46" end="46">.</word>
</words>

<?xml version="1.0"?>
<clause semfeat="nonagentive">
  <phrase synform="NP" synfunc="SB" start="1" end="10">Text books</phrase>
  <phrase synform="VP" synfunc="HD">
    <text start="12" end="16">write</text>
    <phrase synform="NP" synfunc="OA" start="18" end="29">
      this process</phrase>
    <phrase synform="PP" synfunc="MO" start="31" end="45">
      as HCl H+ + Cl-</phrase>
  </phrase>
  <text start="46" end="46">.</text>
</clause>
```

Figure 2: Corpus + aligned annotation layers

```
<?xml version="1.0"?>
<text>
Okay well you turn right and you go along the corridor and you turn
left into the second little corridor and as soon as you do that the
office will be straight on your right
<text>

<?xml version="1.0"?>
<clauses>
  <clause comp="cont" rel="init" start="1" end="24">
    Okay well you turn right</clause>
  <clause comp="cont" rel="para" start="26" end="54">
    and you go along the corridor</clause>
  <clause comp="cont" rel="para" start="56" end="104">
    and you turn left into the second little corridor</clause>
  <clause comp="cont" rel="hypo" start="106" end="131">
    and as soon as you do that</clause>
  <clause comp="fin" start="133" end="173">
    the office will be straight on your right</clause>
<clauses>

<?xml version="1.0"?>
<inton-units>
  <inton-unit tone="t3">
    <text start="1" end="18">Okay well you turn</text>
    <emph start="20" end="24">right</emph>
  </inton-unit>
  <inton-unit tone="t3">
    <text start="26" end="45">and you go along the</text>
    <emph start="47" end="49">cor</emph>
    <text start="50" end="58">ridor and</text>
  </inton-unit>
  <inton-unit>
    <text start="60" end="67">you turn</text>
    <emph start="69" end="72">left</emph>
  </inton-unit>
  <inton-unit tone="t3">
    <text start="74" end="95">into the second little</text>
    <emph start="97" end="99">cor</emph>
    <text start="100" end="104">ridor</text>
  </inton-unit>
  <inton-unit tone="t2">
    <text start="106" end="126">and as soon as you do</text>
    <emph start="128" end="131">that</emph>
  </inton-unit>
  <inton-unit tone="t1">
    <text start="133" end="167">
    the office will be straight on your</text>
    <emph start="169" end="173">right</emph>
  </inton-unit>
</inton-units>
```

Figure 3: Intersecting hierarchies

and McKelvie, 1997), where the granularity of (ordered) reference can go down to a single character (this is useful for example, for marking up word stems or syllables).

Elements are named at the level of specificity needed for validation. For example, when it is necessary to control the context in which SB (subject) or HD (main verb) may occur, the element `<phrase>` needs to be differentiated into elements `<SB>` and `<HD>`. The element `<text>` is used to align text-nodes that occur in mixed content with the source.

Figure 3 demonstrates how intersecting hierarchies from the speech corpus example in Section 2 are represented; for example, the second element `<inton-unit>` overlaps with the third element `<clause>`, but neither one fully contains the other.

3.3 Querying

For both ODAGs represented as XML (McKelvie et al., 2001) and AGs (Bird et al., 2000), query languages have been proposed. In this section we explore how XQuery fares for querying multi-layer corpora.

XQuery (W3C, 2001a) is a query language for XML currently standardized by the W3C. Its surface syntax consists of three main constituents: (1) FLWR-expressions provide `for`-clauses to bind and iterate over variables, `let`-clauses to bind variables, `where`-clauses to filter tuples of bound variables with predicates, and `return`-clauses to specify the result of a query. (2) XPath-expressions (W3C, 1999) select document fragments, and (3) XML syntax can be used to generate results. The underlying semantics of XQuery (W3C, 2001b) is specified by means of a fully compositional, functional, strongly typed language.

By relying on XPath and a rather rich, evolving set of functions and operators, XQuery supports document-oriented queries against a single hierarchy that take into account sequential order and hierarchical structure fairly well (see also (W3C, 2001c)). For querying multiple, possibly intersecting hierarchies, we need in addition two user-defined boolean functions; `my:contains` and `my:overlaps`:

```
define function my:contains (AnyElement? $x, AnyElement? $y): Boolean
{
 (($x/descendant-or-self::*[@start])[1]/@start <=
  ($y/descendant-or-self::*[@start])[1]/@start)
  and
 (($x/descendant-or-self::*[@end])[last()]/@end >=
  ($y/descendant-or-self::*[@end])[last()]/@end)
}

define function my:overlaps (AnyElement? $x, AnyElement? $y): Boolean
{
 (($x/descendant-or-self::*[@start])[1]/@start <=
  ($y/descendant-or-self::*[@end])[last()]/@end)
  and
 (($y/descendant-or-self::*[@start])[1]/@start <=
  ($x/descendant-or-self::*[@end])[last()]/@end)
}
```

An element `$x` contains an element `$y` if `$x/@start <= $y/@start` and `$x/@end >= $y/@end`. Recall that in our representation, elements higher up in the hierarchy "inherit" their `start`- and `end`-attributes from their descendants. This is accomplished by the XPath-expression `($x/descendant-or-self::*[@start])[1]/@start` which returns the `start`-attribute from the first element with a `start`-attribute among its descendants. In a similar way, the end position of an element can be determined. Both are rather costly operations, therefore it makes sense to redundantly store `start`- and `end`-attributes for every element, and to maintain indices for them, e.g., by means of B*-trees.

With respect to a common base corpus, hierarchical dominance implies sequential (temporal) containment but not vice versa. More precisely, for every two elements `$x` and `$y` `$x/$y` (i.e., `$y` is a direct child of `$x`), implies `$x//$y` implies `my:contains($x,$y)` implies `my:overlaps($x,$y)`. These four relations provide the basic toolset to query multi-layer annotated corpora in XQuery.

Query 1: Return all nonagentive clauses, that directly dominate a phrase with syntactic function "subject" and contain a word with part-of-speech "NNx".

```
for $c in document('clauses1.xml')//clause[@semfeat='nonagentive']
where some $p in $c/phrase[@synfunc="SB"] satisfies
      some $w in document('pos.xml')//word[my:contains($p,.)] satisfies
            substring($w/@pos,1,2)='NN'
return $c
```

This query retrieves clauses fulfilling the conditions with respect to the phrase-structure annotation from `clauses1.xml` and filters them further by testing the condition with respect to the part-of-speech annotation in `pos.xml`[9](see Figure 2). Because there will be typically many more words with `@pos = NNx` than words contained in `$p`, a suitable query plan for this query is to retrieve first words contained by `$p`, using indices on `start`- and `end`-attributes.

Query 2: Return all continuing clauses that overlap with an intonation with tone "t3".

```
for $c in document('clauses2.xml')//clause[@comp='cont']
where some $i in document('inton-unit.xml')//inton-unit
      satisfies my:overlaps($c,$i) and @tone='t3'
return $c
```

This query is posed against the annotation layers in Figure 3. It proceeds similarly to Query 1. The overall design pattern for queries over multiple annotation layers is typically: (1) Divide the query into subqueries addressing the conditions for individual annotation layers, (2) combine the results of the subqueries by means of `my:contains` or `my:overlap`. (3) Optionally, reformulate the query such that a moderately clever query-optimizer can devise an efficient query plan.

Compared to the special purpose query languages proposed in (McKelvie et al., 2001) and (Bird et al., 2000), the queries above are certainly significantly more verbose[10]. This reflects the principle tradeoff between simplicity and generality: At the cost of conciseness, we have resisted introducing syntactic extensions for the purpose of querying multiple layers, but rather relied on XQuery's built-in mechanisms for referring to multiple documents and user-defined functions.

In this outline, we have focused on the specific problem of querying multi-layer annotations aligned on a common base corpus. Other useful query classes that we have encountered in our work are: Limited forms of regular expressions to look for syntactic patterns (e.g., passive constructions in part-of-speech annotations), dictionary lookups, and traversal of relationships expressed by shared attributes, but notably not the need for transitive closures of such relationships.

[9] In XPath-filter expressions ([*expr*]), node-sets are implicitly converted to booleans. With this approach the `where`-clause could be formulated more concisely. It is an open issue whether XQuery will adopt such a scheme for `where`-clauses.

[10] Both queries can be expressed even more verbosely in XSLT + XPath.

4 Summary and conclusions

In this paper, we have reviewed the state-of-the-art in multi-layer corpus representation. Our goal has been to come up with a set of requirements on corpus representation motivated by the typical informational needs of linguists (rather than, say, computational linguists; Section 2) and to sketch a principled approach meeting those requirements (Section 3). We do not claim that we have presented a complete picture of all the relevant aspects involved in multi-layer corpus representation and querying—our goal has simply been to contribute to furthering the understanding of the main underlying issues.

We strongly believe that there is an increasing interest in informationally-rich corpora on the part of all kinds of branches of linguistics, but that such corpora do not nearly reflect the complexity of linguistic knowledge we are used to dealing with in linguistic theory. For instance, a "typical" linguist is only rarely interested in simple questions, such as 'give me all the verbs in corpus x'. Linguistic research questions are usually considerably more complex. One class of questions is the one about the interaction of different parts of the language resource, such as the mutual constraints between syntax and semantics, between text structure and syntactic structure, or between the properties of phonological units and grammatical units.

If we want to enrich corpora with multiple kinds of linguistic information, we need nothing less than a linguistically-motivated model of the kinds of linguistic relations we would like to inquire about on the basis of an annotated corpus. So the first and fundamental challenge on the part of linguistics is to provide a classification of the kinds of linguistic relations considered relevant. Some of them have been explicitly acknowledged in corpus annotation and models for representation have been proposed, notably for dominance and dependency (syntax) and for temporality (speech). But there are many more types of relations we have not even begun to address, among them inter-stratal relations (as illustrated by our second example in Section 2: grammar and phonology), agreement relations and other kinds of 'prosodic' phenomena (often represented by feature/structure sharing), syntactic binding relations (typically represented by various tracing mechanisms), cohesive relations (referent-antecedent relations, lexical cohesion, logico-semantic relations, ellipsis and substitution), coherence phenomena (such as rhetorical structure and theme-rheme/given-new distribution). It is here that the challenge lies for the linguist: to provide a comprehensive list of requirements of these relations and their representation.

¿From a technical perspective, multi-layer corpora constitute an interesting special case of information superimposed on existing data (Maier and Delcambre, 1999). The technical challenges for practical and scalable multi-layer corpora are in particular robust and possibly parameterized addressing schemes to superimpose information on corpora, well tuned indexing, factorization, and query processing techniques, and suitable models and techniques to validate superimposed information.

References

M. Baker. 1995. Corpora in translation studies: An overview and some suggestions for future research. *Target*, 7(2).

S. Bird and J. Harrington, editors. 2001. *Speech Communication. Special Issue on Speech Annotation and Corpus Tools*, 33(1-2).

S. Bird and M. Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1-2):23–60.

S. Bird, P. Bunemann, and W.-C. Tan. 2000. Towards a query language for annotation graphs. In *Proceedings of LREC 2000*, Athens.

T. Brants. 2000. TnT —A statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP) 2000*, Seattle, WA.

S. Cassidy and J. Harrington. 2001. Multi-level annotation in the EMU speech database management system. *Speech Communication*, 33(1-2):61–78.

D. Florescu and D. Kossman. 1999. Storing and Querying XML Data using an RDMBS. *IEEE Data Engineering Bulletin*, 22(3).

M.A.K. Halliday. 1967. *Intonation and grammar in British English.* Mouton, The Hague.

S. Hansen and E. Teich. 2001. Multi-layer analysis for translation corpora: methodological issues and practical implications. In *Proceed-*

ings of Workshop on Multi-Layer Corpora, Iasi, Romania.

U. Heid and A. Mengel. 1999. Query language for research in phonetics. In *Proceedings of the International Congress of Phonetic Sciences (ICPhS 99)*, San Francisco.

E. Hinrichs, H. Feldweg, M. Boyle-Hinrichs, and R. Hauser. 1995. Abschlußbericht ELWIS. Korpusunterstützte Entwicklung lexikalischer Wissensbasen für die Computerlinguistik. Technical report, Universität Tübingen, Tübingen.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19.

D. McKelvie, A. Isard, A. Mengel, M.B. Moller, M. Grosse, and M. Klein. 2001. The MATE workbench — An annotation tool for XML coded speech corpora. *Speech Communication*, 33(1-2):97–112.

D. Maier and L. M.L. Delcambre. 1999. Superimposed Information for the Internet. In *Proceedings of ACM SIGMOD Workshop on The Web and Databases (WebDB'99)*, Philadelphia, Pennsylvania.

A. Mengel and W. Lezius. 2000. An XML-based representation format for syntactically annotated corpora. In *Proceedings of LREC 2000*, Athens.

A. Neumann. 1999. Parsing and Querying XML Documents in SML. PhD Thesis, Universität Trier, Trier.

M. O'Donnell. 1995. From corpus to codings: semi-automating the acquisition of linguistic features. In *Proceedings of the AAAI Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford, California.

O. Plaehn and T. Brants. 2000. Interactive corpus annotation. In *Proceedings of LREC 2000*, Athens.

G. Sampson. 1995. *English for the Computer*. Oxford University Press, Oxford.

E. O. Selkirk. 1984. *Phonology and Syntax: The Relation between Sound and Structur*. MIT Press, Cambridge, USA.

C. M. Sperberg-McQueen and C. Huitfeldt. 2001. GODDAG: A Data Structure for Overlapping Hierarchies. In *Proceedings of PODDP'00 and DDEP'00*, New York.

M. Steedman. 2000. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689.

E. Steiner. 2000. Translations English – German: some observations on the relative importance of systemic contrasts and of the text type "translation". Paper presented at the Oslo Symposium on Information Structure in a Cross-linguistic perspective, December 2000. `URL: www.hf.uio.no/german/sprik/english/reports.shtml`.

E. Teich and S. Hansen. 2001. Methods and techniques for a multi-level analysis of multilingual corpora. In *Proceedings of Corpus Linguistics 2001*, Lancaster, UK.

E. Teich, C. I. Watson, and C. Pereira. 2000. Matching a tone-based and a tune-based approach to English intonation for concept-to-speech generation. In *Proceedings of the 18th International Conference on Computational Linguistics (Coling) 2000*, Saarbrücken, Germany.

E. Teich. 2001. Contrast and commonality in English and German system and text. A methodology for the investigation of the contrastive-linguistics properties of translations and multilingually comparable texts. Universität des Saarlandes, Saarbrücken.

H. S. Thompson and D. McKelvie. 1997. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe 97*.

World Wide Web Consortium. 1999. XML Path Language (XPath) Version 1.0 - W3C Recommendation. `URL: www.w3.org/TR/xpath.html`.

World Wide Web Consortium. 2000. Extensible Markup Language (XML) 1.0 (Second Edition) - W3C Recommendation. `URL: www.w3.org/TR/2000/WD-xml-2e-20000814`.

World Wide Web Consortium. 2001a. XQuery: The W3C Query Language for XML - W3C Working Draft. `URL: www.w3.org/TR/xquery/`.

World Wide Web Consortium. 2001b. XQuery 1.0 Formal Semantics - W3C Working Draft. `URL: www.w3.org/TR/query-semantics/`.

World Wide Web Consortium. 2001c. XML Query Use Cases - W3C Working Draft. `URL: www.w3.org/TR/xmlquery-use-cases`.