ANNIS: A Linguistic Database for Exploring Information Structure

Stefanie Dipper[*], *Michael Götze*[*], *Manfred Stede*[*] *and Tillmann Wegst*[†]

[*]Universität Potsdam and [†]Saarbrücken

In this paper, we discuss the design and implementation of our first version of the database 'ANNIS' ('ANNotation of Information Structure'). For research based on empirical data, ANNIS provides a uniform environment for storing this data together with its linguistic annotations. A central database promotes standardized annotation, which facilitates interpretation and comparison of the data. ANNIS is used through a standard web browser and offers tier-based visualization of data and annotations, as well as search facilities that allow for cross-level and cross-sentential queries. The paper motivates the design of the system, characterizes its user interface, and provides an initial technical evaluation of ANNIS with respect to data size and query processing.

1 Introduction

Information structure (IS) is an area of linguistic investigation that has given rise to a multitude of terminologies and theories that are becoming more and more difficult to survey. The basic problem is that IS-related phenomena often can be observed only indirectly on the linguistic surface and hence invite competing interpretations and analyses that are tailored to the needs and the flavours of the participating researchers. Thus, in contrast to syntax, where different approaches can be—more or less—systematically compared, with IS it is often not even clear whether two theories compete to describe the same phenomenon or are in fact complementary to each other, characterizing linguistic regularities on different levels of description.

In 2003, a long-term research infrastructure ('Sonderforschungsbereich', henceforth 'SFB') has been established at Potsdam University and Humboldt-

Interdisciplinary Studies on Information Structure 01 (2004): 245–279
Ishihara, S., M. Schmitz and A. Schwarz (eds.):

University Berlin.[1] Its idea is to investigate the various facets of IS from very different perspectives and to contribute to a broader and more general understanding of the phenomena by bringing the various results together and promoting the active exchange of research hypotheses. Participating projects (see Section 2) provide empirical data analyses that will serve as the basis for formulating theories that aim at advancing the state of the art and overcoming the unpleasant situation characterized above.

An important prerequisite for this long-term and multi-disciplinary approach is the ability to *annotate* the data with appropriate information and to collect the variety of data in a single, uniform database.[2] Given the present situation, annotation sets cannot be presumed to be identical—different researchers will first start out with their own favourite terminology. The convergence of the annotation sets is an important goal for the SFB, and the idea is that this process can be actively promoted by making the interim analyses of the various projects accessible, to invite comparison and possibly revision. Specific working groups dedicated to various levels of analysis are in charge of monitoring this process.

In this paper, we discuss the design and implementation of our first version of the database 'ANNIS' ('ANNotation of Information Structure'). Section 2 provides some more details about the SFB and summarizes the particular requirements that this research scenario places on developing the database. Section 3 explains the architecture, user interface, and query facilities of the current implementation. Then, Section 4 illustrates the operation of ANNIS with an example. Section 5 presents an evaluation of the current state of the database. In Section 6, we compare our approach to related work, and Section 7 discusses our plans for future extensions.

[1] `http://www.ling.uni-potsdam.de/sfb/`

[2] For a comparative evaluation of various annotation tools, see Dipper et al. (2004).

2 The SFB

The SFB consists of 13 individual research projects from disciplines such as theoretical linguistics, psycholinguistics, first and second language acquisition, typology, and historical linguistics. Following the overarching objective of providing a clearer picture of information structure, several of the projects are involved in collecting and analyzing empirical data. Here are some examples of such activities.

Semantics and IS One project examines the relation between *quantifier scope* and IS. Data is annotated with semantic features such as quantifier scope, referent identifiability, and definiteness. Another project investigates interactions between semantic focus evaluation, discourse anaphoricity, and presupposition.

IS and discourse structure One project is interested in the effects that rhetorical relations and discourse structure in general can have on the prosodic structure of spoken discourse. The data to be annotated with corresponding features are radio news broadcasts.

Focus in African languages Two projects examine the phenomenon of focus in different Western African languages. Both carry out field studies for collecting data, which is later being annotated.

Diachronic change One project investigates the evolution of the verb-second phenomenon, which occurred in certain Germanic languages only (e.g., it did in Modern German, but not in Modern English). Based on manuscripts of Old High German and Old English, the role of IS in this evolution will be studied.

Typology of information structure One project seeks to develop a typology of the means for expressing IS. In close cooperation with the other projects, a

questionnaire is being developed that will serve as a basis to collect language data relevant for IS from speakers of typologically diverse languages.

2.1 The data

As pointed out above, the individual projects apply different means in collecting data, and they focus on different aspects of IS. Hence, both the *primary* data (i.e., the language data that is collected) as well as the *secondary* data (i.e., the annotations to the primary data) of these projects differ in several respects.

Primary data The source data can consist of recorded speech, or videos of spoken monologues or dialogues. Furthermore, some projects work with written texts, either in digital form or as original manuscripts. A special case is the above-mentioned *questionnaire*, whose primary data are answers to questions. Generally, the data is taken from diverse languages, many of which do not make use of the Latin character set.

Secondary data Languages differ with respect to the means they exploit to express IS (e.g. stress, word order, particles). Depending on the objectives of the individual project, the secondary data thus relates to phonetic or phonological, morphological, syntactic, or semantic properties. The encoding of these properties requires, e.g., simple attribute-value pairs (e.g. for morphological features), trees (syntax), undirected relations or pointers (co-reference).

Metadata represents another type of secondary data: information that relates to a speech or text sample as a whole and, e.g., encodes the date of recording, information about the speaker or author (sex, age, etc.). Other metadata refers to the language of the sample, in the form of typological information (e.g. ergative language), genealogical information, or areal data.

Finally, the *questionnaire* also represents a kind of annotation. The questionnaire consists of pairs of stimuli (e.g. questions, pictures, or films that are used to trigger speech) and data elicited by these stimuli. These pairs are organized in a hierarchical manner, i.e., there are more general and more specific questions, questions that presuppose other questions, etc.

2.2 Requirements

The general objective for the ANNIS effort is to provide a common database for the data collected and annotated by the individual projects. This database has to serve as long-term data storage and, at the same time, offer convenient access to the data, through search facilities and a graphical user interface for display. The research scenario characterized above places different types of demands on this database, which we briefly describe in this section.

Standard formats In order to promote convergence of the annotations performed by different projects and researchers, a common standardized annotation format is of great importance. Therefore, SFB-wide working groups are defining an *SFB Annotation Standard* with tagsets and annotation guidelines for morpho-syntax, prosody, semantics/pragmatics, and information structure.

Moreover, we are developing a common standardized representation format, the *SFB Encoding Standard*. This format represents the data and their annotations in a uniform way and allows for stating constraints on the content of the annotation. It thus facilitates the comparison of different tagsets (Which tags are used by all projects? Which tag occurs in one type of data only? Etc.). Moreover, it allows for consistency checks (only predefined tags are allowed) and completeness checks (certain annotation levels are to be annotated obligatorily).[3]

[3] The SFB *Annotation* Standard defines the *tag sets* that constrain the 'content' of the sec-

Further, the data of the SFB will step by step be made available to the research community. To facilitate data exchange and reuse, world-wide standard formats have to be supported: import and export format of the database must be based on XML, which allows for data exploitation and manipulation by many existing programs and tools—both general-purpose and linguistic tools, such as search tools, annotation tools, converters, and databases. Moreover, the import and export format should comply with standardized linguistic XML applications, i.e. specifications for XML-based representations of linguistic features (e.g. TEI[4], XCES[5]).

Flexibility As mentioned above, primary as well as secondary data of the projects differ to a large extent. The database has to be sufficiently flexible to accommodate the different kinds of data. At the same time, the database should adapt to the specific needs of individual projects. For instance, sometimes intra-sentential and inter-sentential (discourse) annotation are to be combined. Hence, the database has to provide suitable visualization of both intra-sentential annotation (such as syntactic trees) and inter-sentential annotation (e.g. co-reference relations).

Querying As studying information structure involves relating different types of information—and hence annotation—, it is important that queries to the database can easily span across different levels of annotation. Furthermore, it is important to be able to restrict the scope of queries, so that a researcher can search, for example, only the data collected by her/himself, or that assembled by a particular project, or data of a specific genre (such as spoken dialogue).

ondary data. The SFB *Encoding* Standard determines the format of the internal representation of primary and secondary data.

[4] `http://www.tei-c.org/P4X/`

[5] `http://www.cs.vassar.edu/XCES/`

Modeling of the questionnaire The database should model the structure of the questionnaire. For instance, it should allow the user to navigate from general to specific questions, to navigate from a question-answer pair in language X to the corresponding pair in language Y (whose data has been elicited on the basis of the same questionnaire).

Operability The database should be easy to operate. It should support straightforward retrieval of linguistic phenomena and an intuitive display of the primary and secondary data, so that linguists who are not experts in using databases can profit from the endeavour.

2.3 Application scenarios

The database has to be designed in such a way that it supports two rather different application scenarios. The first, henceforth called ‘scenario A’, is that of a centralized data repository for the SFB and beyond. Via the WWW, the data is to be made accessible to interested parties. The second, ‘scenario B’ is the role as research vehicle within an individual project: Data that has just been collected is annotated—maybe in a first pass rather than ‘final’—and checked for consistency; first hypotheses are to be tested, which might lead to changes in the annotation; gaps in the annotation tag set might be identified. This kind of work has a clearly local, premature character and should not necessarily be executed on the ‘official’ central database. Instead, the system should also run on a local PC or laptop, where the projects can prepare their data until it has reached a state allowing for sharing it with others.

3 The Database

The requirements just outlined motivated the basic design decisions for the database system. In the following, we first explain its overall architecture in

somewhat more technical terms (Section 3.1).[6] Then, Section 3.2 introduces some terminology to be used in the subsequent description of the user interface (Section 3.3) and the query facility for searching data (Section 3.4).

3.1 Architecture

ANNIS is a web application that is accessed with standard web browsers. Technically, at the heart of processing are a Java servlet (which keeps all the data in memory), an open number of XML files providing the data, plus a number of DTDs, configuration files, and resources.

In addition to the requirements from the perspective of the linguistics researcher, there are a number of technical factors influencing the design. ANNIS should be:

- widely and easily accessible,
- fast with regard to display and searching,
- open with regard to integration of data from heterogeneous sources and, at the same time, supportive of our aim to create a standard format,
- open with regard to passing data on to external applications and uses,
- portable across the boundaries of operating systems, and
- configurable with regard to interface language and look and feel.

In order to comply with these goals, ANNIS was designed around the following main decisions.

[6] Readers who are not interested too much in technical details might want to skip this section.

Web-based Being a web application, ANNIS fulfills the criterion of universal, easy accessibility. Prerequisites on the client side are modest, as (for the most part) no special plug-ins are required. Instead, the implementation uses only HTML, CSS and JavaScript.

RAM based ANNIS is a database-backed web application. Standard usage of the term 'database' is somewhat misleading, however, since there is no genuine DBMS being used. Rather, the application reads its data from files at startup and keeps them completely in memory during a session. This was motivated by the criterion of speed; in particular, query execution profits a lot from ANNIS being memory-based. The ANNIS query language allows the construction of complex queries, employing regular expressions, grouping, disjunction, conjunction, negation, constraints on relations between nodes within trees, etc., which for an SQL processor would be expensive to analyse and execute, memory-consuming in the case of complex joins, and therefore running rather slow.

A potential reason for using a DBMS might be the ease with which data can be added, changed and deleted at runtime. However, in our application scenario A (with a centralized data repository, cf. Section 2.3), the data will be relatively stable (annotators move it from their PC to the main database only when the work is considered finished). Still, to keep track of changes, ANNIS provides an incremental update component that detects added, modified and missing files and updates the data in memory accordingly. In application scenario B (with local installation), where data change is indeed an issue, the local database can be expected to be quite small so that speed problems are very unlikely.

Dynamic importer plugin At present, data formatted according to seven different XML document type definitions (*inter alia*, stemming from the annota-

tion tools EXMARaLDA[7] and MMAX[8], and the TIGER[9] syntax annotations) can be imported into the ANNIS system. Since formats are undergoing changes and new formats are entering the scene, special care was taken to ease the process of integrating new importers. Even though we consider the development of a common XML format as an important objective for the SFB (see above), import facilities nonetheless play an important role when ANNIS is distributed to other interested parties. Therefore, ANNIS was built in such a way that adding or replacing a data importer requires no recompilation of the system as a whole. It suffices to add the new or modified Java class side by side to the other classes making up the system. It is even possible to do so in the midst of an ANNIS session: importers can be plugged in at runtime.

Export and conversion ANNIS provides several ways to export data, allowing for inspecting the data in its XML form and for externally using it in other applications. In particular, the XML data may be shown in the browser (optionally converted to an HTML representation of the data), downloaded, or sent to an email address, or deposited in a directory on the ANNIS server, optionally zip compressed.

Data can be exported in its original format, or be converted to the SFB Encoding Standard format, which we are developing as a general representation that abstracts over the peculiarities of the various annotation tool formats. At the moment, though, the SFB standard format can only be imported to ANNIS; the export module will follow.

Pure Java The use of pure Java for all server-side machinery allows ANNIS to run on all platforms providing a Java virtual machine and a Java servlet en-

[7] `http://www.rrz.uni-hamburg.de/exmaralda/`
[8] `http://www.eml-research.de/english/research/nlp/download/index.php`
[9] `http://www.ims.uni-stuttgart.de/projekte/TIGER/`

gine. So far, ANNIS has been installed on Windows NT, Windows XP, Mac OS X and Linux, in each case under an Apache Tomcat web server running in standalone mode.

Localization and adaptation At present, the user interface can be configured to run in English or German mode. The localization for other languages would pose no problem. Users may adapt the appearance of ANNIS in a number of ways, e.g. with regard to screen size, tool tips, and the like. Administrators can in addition control colors and other elements of style.

3.2 Concepts and notions

In the next sections, we address visualization and querying of data within ANNIS. To ease reference to the data and concepts of ANNIS, we now introduce some notions and illustrate them by example annotations. The example text is annotated by part of speech (POS), cognitive status (COGN-ST), topichood (TOPIC), see Figure 1, and syntax, see Figure 2. The tags used there will be explained below.

Primary data Primary data is the source data, i.e. the text (or speech) that is to be annotated by linguistic data. The primary data in the example in Figure 1 is *Eier-Produzenten aus der ganzen Republik machen . . .* (‘Egg producers from all over the republic make . . . ’).

Secondary data Secondary data consists of the linguistic data that is attached to primary data. For instance, the part-of-speech annotation in Figure 1 represents secondary data.

TOPIC	aboutness-topic						
COGN-ST	inferrable						
POS	NN	APPR	ART	ADJA	NN	VVFIN	
Text	Eier-Produzenten	aus	der	ganzen	Republik	machen	...

Figure 1: Example annotation, encoding topichood, cognitive status, and part of speech

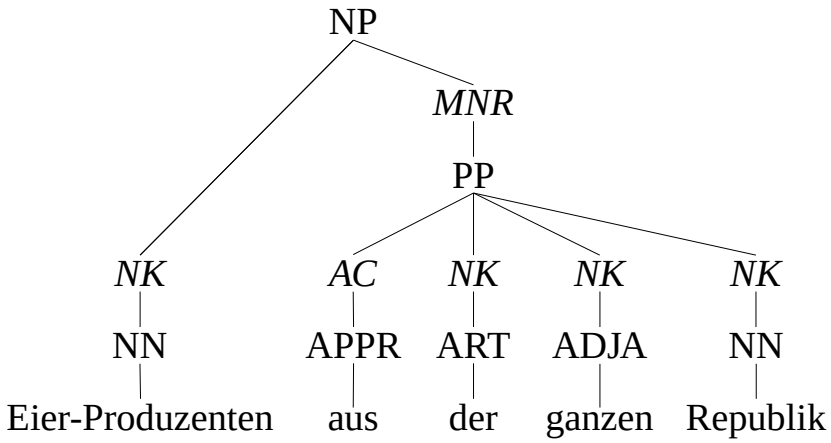

Figure 2: Example annotation, encoding part of speech, syntactic categories, and grammatical functions (TIGER syntax; functions are set in italics)

Annotation level Annotations are grouped according to linguistic domains, which correspond to annotation levels, e.g. part-of-speech or information-structural annotation levels.

Complex linguistic domains may be broken into smaller levels. For instance, information-structural properties can be represented by different annotation levels, such as cognitive status and topichood as in Figure 1.

Competing analyses of the same domain are considered distinct annotation levels. For instance, there can be an STTS[10] POS annotation level (i.e., the analyses comply with the STTS annotation guidelines) vs. an SFB POS

[10] `http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-1999.ps.gz`

annotation level (with analyses according to the SFB Annotation Standard for part-of-speech annotation).

Each annotation level is characterized by a specific tagset.

Tagset A tagset is the set of attribute-value pairs (= tags) that are admissible at a specific annotation level. For instance, the part-of-speech annotation level can specify STTS as its tagset. STTS makes use of only one attribute, "pos", with 51 different values: "NN" marks common nouns, "APPR" prepositions, etc. Accordingly, an STTS-compliant attribute-value pair is "pos=NN".

Syntactic tagsets often use two attributes, "cat", which encodes the syntactic category, and "func", encoding the grammatical function. Admissible values for the attribute "cat" might be "NP", "PP", etc., and "NK" (noun kernel), "MNR" (modifier of a noun, postnominal ('right')), "AC" (adpositional case marker) for the attribute "func", cf. Figure 2.

Tag An attribute-value pair is called 'tag', e.g. "pos=NN", "cat=NP".

(Atomic) annotation These are the elementary units of any annotation. An atomic annotation consists of a tag that is attached to a segment, i.e. to a piece of primary data (e.g. text) or secondary data (a sequence of atomic annotations).

(i) An atomic annotation can consist of an attribute-value pair that is attached to a piece of primary data. For instance, the annotation "pos=NN" in Figure 1 is attached to the token *Eier-Produzenten*, the annotation "cognitive-status=inferrable" is attached to a sequence of tokens, *Eier-Produzenten aus der ganzen Republik*.[11] Put differently, "pos=NN" is one of the atomic annotations

[11] Technically speaking, part-of-speech annotations are not attached directly to primary data in our implementation. We define characters as the basic units, i.e., atomic annotations of type "char" mark single characters. Next, atomic annotations of type "tok" refer to the basic "char" annotations. "pos" annotations are then attached to "tok" annotations; "pos" (and "tok") annotations are therefore atomic annotations of type (ii) rather than (i).

TOPIC	aboutness-topic						
	is-domain						
Text	Eier-Produzenten	aus	der	ganzen	Republik	machen	...

Figure 3: Example annotation, encoding the annotation level of topichood (TOPIC) by two attributes displayed on two tiers

of *Eier-Produzenten*, and "cognitive-status=inferrable" is one of the atomic annotations of *Eier-Produzenten aus der ganzen Republik*.

(ii) An atomic annotation can consist of an attribute-value pair that is (recursively) attached to one or more atomic annotations (this is needed for the encoding of hierarchical structures such as trees). For instance, the atomic annotation "func=NK" in Figure 2 is attached to the atomic annotation "pos=NN". The atomic annotation "cat=NP" is attached to a sequence of atomic annotations, "func=NK" and "func=MNR".

Segment A segment defines a sequence of primary or secondary data: a piece of text (a sequence of characters or tokens), or a sequence of atomic annotations.

Instantiated annotation level The set of all atomic annotations belonging to an annotation level is called 'instantiated annotation level'. That is, an instantiated annotation level consists of all attribute-value pairs that are actually used in the annotation—as opposed to the tagset, which defines the range of all attribute-value pairs that could be used.

Annotation layer An annotation layer is the graphical display of an instantiated annotation level. One annotation layer consists of one or more tiers that are stacked on top of each other. For instance, the annotation level of topichood might define two attributes: "aboutness-topic" and "is-domain", which marks general information-structural domains. The segments annotated by these at-

tributes always overlap, since any topic expression must be located within an information-structural domain. Hence, the display of the atomic annotations is spread over two tiers—one displaying the attribute-value pairs of "aboutness-topic", the other displaying "is-domain"—to make the extensions of the segments transparent, cf. Figure 3.[12]

Tier A tier is part of an annotation layer: one line, displaying atomic annotations.

Document A document consists of primary data plus all instantiated annotation levels that refer to this data. In our examples in Figures 1 and 2 (which are based on the same text), the text and the instantiated annotation levels of part of speech, cognitive status, topichood, and syntax form a document.

Corpus A set of documents is a corpus. Corpora can be defined according to criteria such as 'documents with the same object language', 'documents annotated with TIGER syntax', 'documents of the SFB project X', etc.

3.3 Visualization

3.3.1 Tier model

The basic metaphor of visualizing the annotated data in ANNIS is that of a *tier set*. The data window thus consists of a single line of primary text at the bottom, and a variety of annotation layers on top of it. For illustration, Figure 5 below provides a screenshot. Each annotation layer can use its own segmentation of the primary text (with the character being the minimal unit). Browsing through the text for the user means 'horizontal scrolling', for which ANNIS supplies

[12] Instead of distributing the information over multiple tiers, other visual means can be exploited, e.g. bubbles emerging on mouse-over; see Section 4.1.

functions to move the text (and its annotations) forward or backward, character-wise, or in jumps with adjustable lengths. This display mode largely mirrors that of tier-based annotation tools such as EXMARaLDA[13] or Praat[14], and users who are experienced with such tools should get used to ANNIS quite quickly. In addition to the annotation window, the complete source text is displayed at the top of the page, with the portion currently shown in the main annotation window being underlined, so that the current position in the complete document is always transparent.

3.3.2 The role of trees

Opting for a tier-based mode to structure and display the data entails that trees are not the primary vehicle for conveying information. Trees can of course be shown in tiers, but this is not the most natural way to present them (cf. Figure 7 and the discussion in Section 4.3.2). The decision to center the data around a tier-model rather than a tree model followed from the primary purpose of the project: Investigating information structure by seeking correlations between quite different kinds of annotations is easier when the annotation and its visualization makes as little a commitment on structure as necessary—and tiers are the most versatile scheme in this respect.

However, ANNIS offers the possibility to associate images with database entries, in which case a hyperlink is given as part of the data. Pre-stored images of tree structures can be accessed this way, for instance using SVG-files that can be exported from TIGERSearch[15] and displayed in the web browser by the Adobe SVG interpreter. In the same fashion, sound files can be added to the data.

[13] `http://www.rrz.uni-hamburg.de/exmaralda/`
[14] `http://www.praat.org/`
[15] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/`

3.3.3 Displaying multiple layers

The most interesting data for the purposes of the SFB is data annotated with different types of information. For example, the *PCC10* annotations, which will be explained in Section 4, use six annotation levels. When inspecting the data, not each layer will be of relevance for each purpose. Thus, layers can be clicked away individually so that the users can focus their attention on the type of information that is currently of interest.

When the labels of atomic annotations shown in annotation layers have to be shortened (to fit the size of the unit), the full version of the label automatically appears on mouse-over. Similarly, when viewing the tagset, extended explanations can be shown on mouse-over. These and some other features can however be configured by the user (whether they appear on mouse-over or on click, what is the window size, etc.).

3.4 Querying

Similar to visualization, querying in ANNIS is both flexible and adaptable to specific needs. It offers a rich set of search operators that can be applied to different types of data: (i) primary data (text), (ii) secondary data (annotations), and (iii) corpora (collections of annotated texts).

Text searches refer to the surface string (or the transcription of speech); for instance, one can search for specific words (e.g. *erst* ‘only’). Annotations can be searched for attributes (e.g. “topic”) or attribute-value pairs, including relations and pointers. Queries for corpora usually occur in combination with text or annotation queries. They allow the user to narrow down the search space by specifying an individual document or a set of documents. For instance, the query can be restricted to documents of a specific SFB project.

3.4.1 Search expressions

Wildcards, regular expressions Basic searches relate to one word (e.g. `erst` 'only')[16] or to one atomic annotation (e.g. `cognitive-status=inferrable`). These search expressions can make use of wildcards, i.e. special characters that match any character in the string comparison. For instance, `pos=N*` matches both expressions marked as "pos=NN" (common nouns) and those marked as "pos=NE" (proper nouns). Text queries may even use regular expressions: `sag(en?|st|t)` matches surface forms like *sage* or *sagst*.

Cross-level queries Often, queries refer to atomic annotations on different levels, e.g. in a search for an expression that is both annotated as the subject and as being inferrable. Such restrictions can be freely combined by means of the Boolean expression "&": `function=subject & cognitive-status=inferrable`. In ANNIS, these restrictions are evaluated with respect to the text that is annotated by the respective attributes. The query example is then interpreted as follows: any piece of text marked as a subject satisfies the restriction of the first conjunct, and any piece of text marked as being inferrable satisfies the second part. Combining both conditions means in ANNIS: looking for text fragments (within the text pieces) that satisfy both conjuncts simultaneously. That is, the text pieces satisfying the first and second conjunct must overlap and the overlapping part qualifies as a match.[17]

For instance, an annotation might mark an NP as the subject; suppose the NP contains an attributive adjective that is marked by contrastive focus, as illustrated in Figure 4. In this annotation, the adjective fulfills both constraints

[16] In this section, expressions in `typewriter` denote actual query expressions that can be typed into ANNIS. Some of the examples are slightly simplified, though.

[17] Technically speaking, the annotations that satisfy the conjuncts are part of the actual match as well.

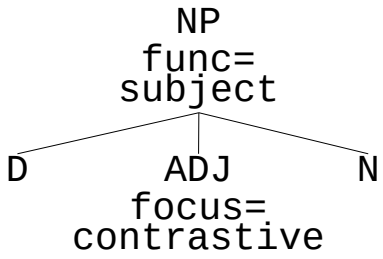

Figure 4: Focus-marked adjective within a subject NP

(being (part of) a subject and being marked by contrastive focus) and, hence, counts as a match to the query.

A stronger restriction requires that text pieces satisfying the conjuncts be identical: `function=subject _=_ focus=contrastive`. Here, the complete subject NP would have to be marked by contrastive focus.

Complex conditions The above examples illustrate the combination of restrictions by means of "&" to form complex queries. Other types of complex conditions are conditions connected by logical "|" ('or'), negated conditions, and conditions on precedence relations between expressions (e.g. an expression marked as inactive which precedes an expression marked as active: `cognitive-status=inactive .* cognitive-status=active`).

Queries for annotations in the form of a tree (e.g. syntax) can in addition refer to dominance relations, node arity (number of children), and left and right corners. For instance, `cat=NP >* cat=PP` searches for NPs that dominate PPs.

Queries across corpora As explained above, queries are evaluated by reference to the text. This means that all annotations of one text can be referenced simultaneously, even if the annotations come from different projects and are

physically part of different corpora (assuming that the same text has been annotated in different corpora).

However, queries may be restricted to documents belonging to a specific corpus, by conditions on the document names: `pcc10*::cognitive-status=inferrable` searches for expressions marked as inferrable in any of the documents belonging to the corpus *PCC10* (see Section 4), i.e. *pcc10.coreference*, *pcc10.is.aboutness-topic*, etc.

3.4.2 Result display

Query results are delivered as a list of hits, each showing the name of the document containing the match, the exact location of the match and the text involved in the match. Documents on this list can be selected and are then displayed with the matching data (text and/or annotations) highlighted. The size of the context to be displayed along with the match can be configured by the user.

3.4.3 History, hit memory, and export

For every user, a history of the queries s/he issued is kept across ANNIS sessions. In addition, users may save selected hits in their personal hit memory, allowing search results to be revisited at a later time.

Matching documents can be exported. However, the export format of the current version of ANNIS does not record the labels specifying those parts of the data that actually matched the query.

4 Example: ANNIS in Action

In this section, we illustrate the operation of ANNIS with the example of the *Potsdam Commentary Corpus* (Stede, 2004), a set of newspaper commentaries that are being annotated on six different levels. In particular, we refer to *PCC10*,

a subset of ten commentaries, for which this annotation has been completed. PCC10 is annotated on the following levels:

(i) co-reference and bridging phenomena, annotated according to the guidelines proposed by Gross (2003),

(ii) information structure with aboutness topics, information focus (or 'rheme') and cognitive status (Götze, 2003),

(iii) part of speech,[18]

(iv) rhetorical relations according to RST (Mann and Thompson, 1988),

(v) connectives (Stede and Heintze, 2004), and

(vi) syntactic structure according to the TIGER treebank format (Brants et al., 2002).

4.1 Data exploration

Figure 5 shows the ANNIS user interface. The menu bar on the left is permanently visible and provides quick access to the most important functionalities of ANNIS, with a search window allowing for formulating corpus queries and navigating in the query history. The workspace on the right is the 'dynamic' part of ANNIS and is used for the various navigation and visualization tasks—for instance for the inspection of the annotation of a document in PCC10.

Our annotation view consists of three components, a *navigation bar* and a *discourse view* at the top, and a *detailed annotation view* at the center.

The detailed annotation view contains a reference line with the textual representation of the primary data at the bottom and the annotations organized

[18] The part-of-speech annotation has been performed by the TnT tagger using the German model, see `http://www.coli.uni-sb.de/~thorsten/tnt/`

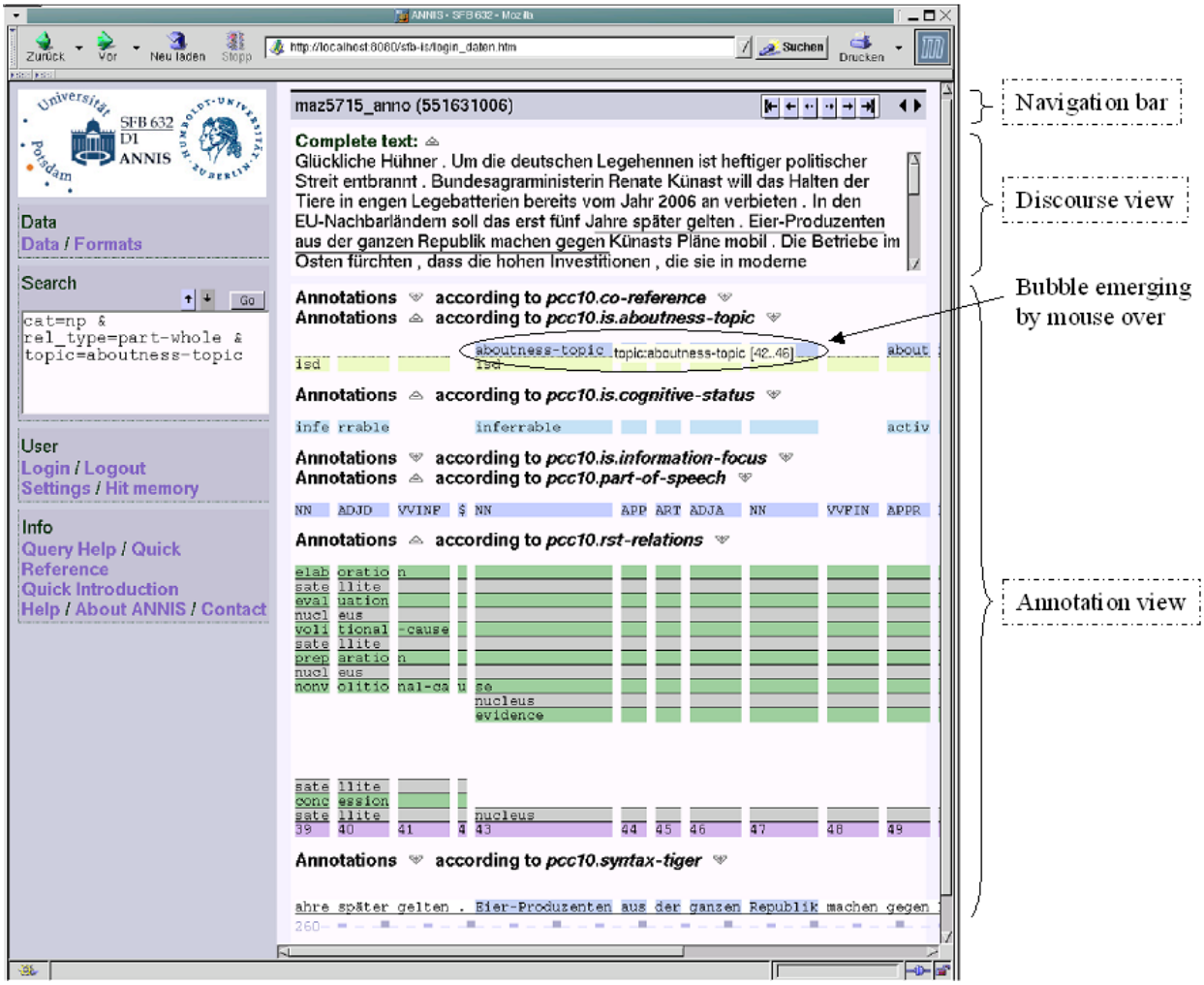

Figure 5: ANNIS user interface

according to the annotation levels above it. In our example, annotations of the levels *pcc10.is.aboutness-topic*, *pcc10.is.cognitive-status*, *pcc10.part-of-speech* and *pcc10.rst-relations* can simultaneously be explored; other, less relevant levels (*pcc10.co-reference*, *pcc10.is.information-focus*, and *pcc10.syntax-tiger*) are clicked away by means of the triangle buttons at each annotation layer.

Annotations are best inspected by moving the mouse over the annotation at the annotation tier: this causes highlighting the primary data associated to it

at the reference line. In Figure 5, the mouse is positioned over the "aboutness-topic" in the upper center, causing *Eier-Produzenten aus der ganzen Republik* to be marked. If the mouse pauses for some time over an annotation, a bubble with more detailed information shows up, in our case displaying its tag ("topic=aboutness-topic") and the numbers representing the span of associated tokens ("42..46").

The discourse view at the top helps users to integrate the data of the detailed annotation view into the larger discourse context. The data currently focused on is underlined. By clicking on a token in the discourse view, the user can shift the annotation view so that this token appears in the center.

By means of the arrow buttons in the navigation bar, we can move back and forth in the data. We may also browse through the documents in the database by the triangular arrows (to the right of the arrow buttons).

4.2 Querying

The search window in the menu bar in Figure 5 contains a multi-level query: `cat=NP & rel_type=part-whole & topic=aboutness-topic`. This expression searches for a nominal phrase ("NP"), whose referent stands in a "part-whole" relation to a previously introduced discourse entity and constitutes an "aboutness topic". After clicking the "Go"-button, ANNIS processes the query and delivers a list of the query results. Figure 6 shows one of the results of the query. In this representation, all of the matching annotation expressions in the query are marked by underlining, i.e. "part-whole", "aboutness topic" and "NP". Again, only annotation levels specified in the query are opened up. An additional button in the navigation bar allows the user to save the result for later inspection.

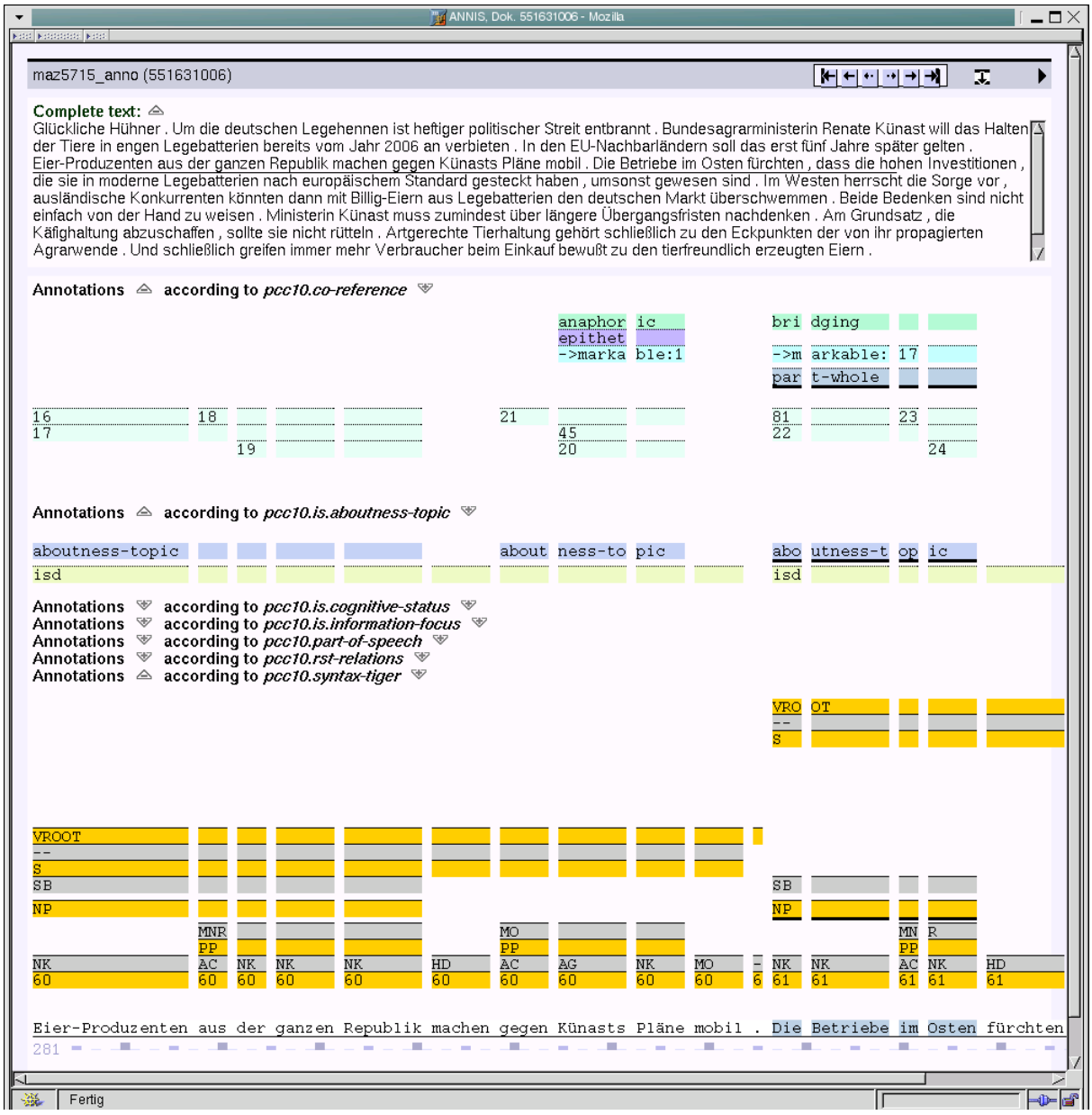

Figure 6: Query result

4.3 Visualization of complex data structures

Figure 6 gives us the opportunity to consider the visualization of two further data types, pointer relations and tree structures.

4.3.1 Pointers

Immediately above the annotation tier with the underlined annotation "part-whole", a pointer annotation is shown: "-> markable: 17". This specifies a pointer relation to the annotation of a tag "markable: 17" at the very left of the *pcc10.co-reference* annotation level in Figure 6.[19] Thus, the referent of *Die Betriebe im Osten* ('The factories in the east') stands in a "part-whole" relation to the referent of the expression marked by "markable: 17": *Eier-Produzenten aus der ganzen Republik* ('Egg producers from all over the republic') in the preceding sentence.

Due to the limited size of the data segment that can be inspected in the annotation view, the current visualization is of limited use, above all for pointer relations crossing larger spans of discourse. We therefore plan to extend the functionality of the discourse view with an improved visualization of pointer relations.

4.3.2 Tree structures

In Figure 7, the tier-based representation of trees in ANNIS can be compared to conventional tree representation. The upper part reproduces a small portion of the syntactic annotation of Figure 6, and the lower part shows the corresponding tree.

[19] The segment that is annotated by the tag "markable: 17" only displays the number "17" in the annotation level of *pcc10.co-reference*. This segment spans the text fragment *Eier-Produzenten aus der ganzen Republik*.

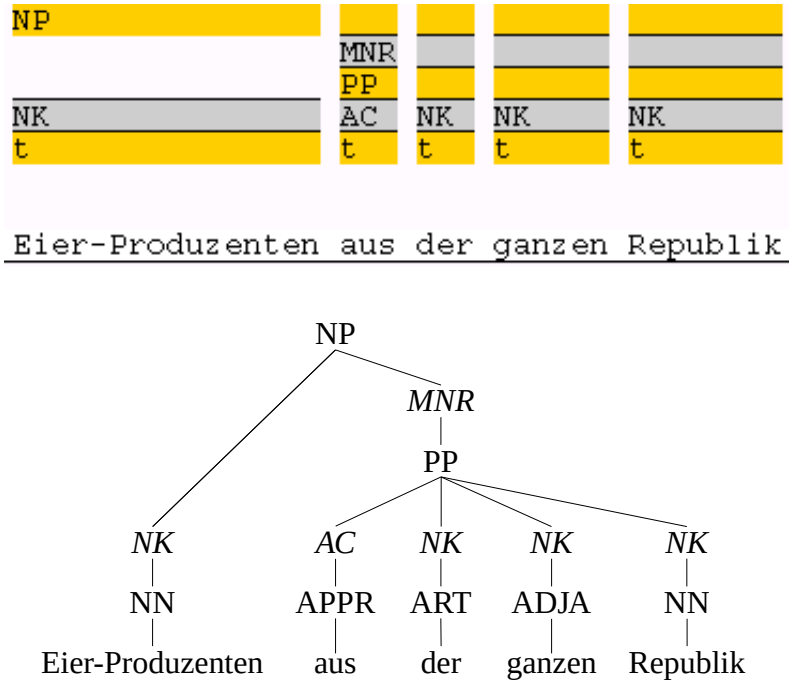

Figure 7: Syntactic annotation represented by ANNIS tiers vs. a tree

Starting from the topmost tier, tiers representing tree nodes (syntactic categories) and those representing tree edges (grammatical functions) alternate. The node “NP” directly dominates the pre-terminal of *Eier-Produzenten* via an edge “NK” (for noun kernel modifier) and the node of cat “PP” (prepositional phrase) via an edge “MNR” (for postnominal modifier). “PP” in turn directly dominates the pre-terminals of *aus der ganzen Republik* via edge labels “AC” (adpositional case marker) and “NK”, respectively.

5 Evaluation

We tested the prototype in the two application scenarios presented in Section 2.3. In scenario A, ANNIS offers its services via the internet, running on a web server with a Pentium IV 2,4 GHz CPU and 3 GB memory. In scenario B, the application is run in standalone mode on a single computer, typical for

linguists wishing to work independently, for instance during field studies. We used a mobile computer with rather low hardware capabilities (Pentium III 650 MHz CPU, 256 MB RAM) for this scenario.

During the evaluation, we focused on two very general aspects: (i) the amount of data that can simultaneously be loaded into ANNIS and (ii) the querying capabilities.

5.1 Data

Due to the RAM-based approach, the amount of data that can be loaded into ANNIS depends on the memory capacities of the hosting machine. While the whole TIGER Corpus (Brants et al., 2002) with more than 40.000 syntactically annotated newspaper sentences can be loaded onto the web server (scenario A), the 1.4 GB of RAM required for this go beyond the capacities of the hardware in scenario B.

We therefore designed two data sets—L(arge) and S(mall)—for this evaluation. Both contain the Potsdam Commentary Corpus of 173 RST-annotated newspaper commentaries and the richly annotated subset PCC10. The sets differ with respect to the number of TIGER sentences they include: the former comprises the whole TIGER Corpus, the latter a subset of 1.000 sentences. Thus, data set L contains approximately 42.200 sentences, and data set S contains 3.200 sentences.[20]

With data set S, the upper limit of the amount of data fed into ANNIS is reached for the laptop. On the web server, data set L occupies 1.4 GB of RAM—even here, the limits of the hardware become relevant.

[20] This results in the following number of Java annotation objects in ANNIS: for L(arge): 3.369.930 objects, for S(mall): 146.505 objects.

5.2 Querying

In addition to flexible query facilities (cf. Section 3.4), ANNIS aims at providing a fast search. Beside keeping the data to be searched completely in memory, ANNIS includes running searching and result delivery in parallel threads: whenever a document is finished with, results found in it are immediately sent to the client—the user can explore the results while more results are still searched for. We therefore measured both the time until the emergence of a first result and the time needed for providing the complete list of results.

Data sets Since data set L cannot be loaded onto the laptop, it was queried on the web server only. Data set S was tested both on the server and the laptop, enabling statements about the performance behaviour depending on the corpus size.

Example queries and evaluation method A small set of queries of different complexity was designed: Query Q1 queries anaphoric expressions as simple attribute-value pairs; Q2 searches for expressions marked as "anaphoric" and "subject", searching across different annotation levels. Finally, Q3 exemplifies a query on hierarchical structures: it queries sentences with a subject nominal phrase that directly dominates a prepositional phrase. [21]

The queries were posed in standard web browsers and the time needed for presenting a first result and the complete list of matches was taken. Thus we did not measure the performance of the search engine alone, but the performance of ANNIS as a whole, including the construction of an HTML representation of the hit list.

[21] The queries have the following form:
Q1: `rel_type=anaphoric`
Q2: `rel_type=anaphoric & rel=SB & #1_=_#2`
Q3: `cat=S & cat=NP & cat=PP & #1>SB#2 & #2>#3`

Scenario	Query	First match (in sec.)	Completed Search (in sec.)	Hits
A (web-server)	Q1	0.5	0.5	70
	Q2	0.5	0.5	5
	Q3	2	2	130
B (mobile computer)	Q1	4	17	70
	Q2	18	18	5
	Q3	20	91	130

Figure 8: Query performance with data set S(mall)

Scenario	Query	First match (in sec.)	Completed Search (in sec.)	Hits
A (web-server)	Q1	5	8	70
	Q2	8	8	5
	Q3	2	34	4985

Figure 9: Query performance with data set L(arge)

Results and discussion The results for querying data set S and L (given in Figures 8 and 9, respectively) show that the overall performance of the ANNIS prototype has still to be improved, particularly with respect to research scenario B, the mobile computer. Here, more complex queries such as Q3 require unacceptable processing times.

However, the strategy of an incremental presentation of query results pays off: with both data sets the first match for Q1 and Q3 is given rather quickly, even if the complete search is time-consuming.[22]

The results also illustrate the expected fact that performance of ANNIS is dependent (i) on the size of the corpus and (ii) the hardware capabilities. On the web server, queries Q1 and Q2 need considerably more processing time with the data set L than with set S. Figure 8 illustrates the difference between

[22] Results are currently presented document-wise. Since all hits of Q2 are part of the same document, the values ‘First match’ and ‘Completed search’ do not differ.

both research scenarios (with different hardware conditions) regarding the query performance: even the processing time for the simple query Q1 differs considerably.

Of course, these first results of a rather shallow testing cannot substitute for an in-depth study of the querying capabilities of the ANNIS search engine, which is planned to be undertaken in the near future.

6 Related Work

Current corpus exploration and query tools do not fulfill all of the needs of the SFB, as presented in Section 2. In this section, we discuss a selection of tools, concentrating on (i) web-based interfaces and (ii) query tools for complex, richly annotated data, and show how they relate to ANNIS.

6.1 Web-based interfaces

Web-based interfaces provide the quickest and easiest access to large amounts of language data and are invaluable tools for linguistic research based on corpora. Simple search facilities allow for querying the data, which usually consists of tokenized text, rarely accompanied by further levels of annotation such as part-of-speech or lemma. Search results are usually presented as plain text or as key word in context, *KWIC*. Prototypical examples are COSMAS II[23] and the online web demos of *Digitales Wörterbuch der deutschen Sprache*[24] and BNC[25].

A tool that is similar to ANNIS by providing access to very heterogeneous data and annotations is the interface of TUSNELDA (‘Tübingen collection of

[23] IDS Mannheim, `http://www.ids-mannheim.de/cosmas2/`

[24] Berlin-Brandenburgische Akademie der Wissenschaften, `http://www.dwds.de/pages/pages_woebu/dwds_woebu_rech.htm`

[25] British National Corpus, `http://sara.natcorp.ox.ac.uk/lookup.html`

reusable, empirical, linguistic data structures')[26]. Beside searching for pure text, TUSNELDA allows the user to specify complex queries in the standardized query language XML QUERY (XQUERY)[27], which is applied to the XML-based annotations. The results of a query are shown as text (for text searches) or as XML representations (for queries on annotations). Display of the XML encoding suffices in many cases, since TUSNELDA annotations rarely cover more than one annotation level—in contrast to our research scenario.

Using XML QUERY has several advantages: Being a standardized language, it is already familiar to at least some users; the format is supported by other tools; and it is a very powerful language. Of course, using XML QUERY requires knowledge of the XML encoding of the annotation.

6.2 Query tools for complex data

In recent years, a number of tools that allow for querying and visualizing more complex annotations have been developed. These include tools for querying trees or graphs and search tools for corpora with multi-level annotation.

Trees/graphs Examples of tree and graph query tools are VIQTORYA (Steiner and Kallmeyer, 2002), TIGERSearch[28], and Netgraph[29]. These tools enable the user to query hierarchical structures and complex relations. Moreover, they include graphical interfaces to improve operability by non-experts and casual users. These interfaces allow the user to compose a query by mouse clicks and simple menu choices. For instance, attribute-value specifications can be selected from a menu which lists all admissible attribute-value pairs. Query results are visualized as trees or graphs.

[26] http://www.sfb441.uni-tuebingen.de/tusnelda-online.html
[27] http://www.w3.org/XML/Query
[28] http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/
[29] http://quest.ms.mff.cuni.cz/netgraph/

However, these tools focus on sentence-based annotations of syntactic structures. That is, inter-sentential queries cannot be posed, and conflicting hierarchies (such as diverging segmentation of primary data by different annotation levels) are not accounted for.

Multi-level annotation A tool that was developed for multi-level annotation is NXTSearch[30]. It is a highly flexible tool in that it can be applied both to time-aligned and hierarchical corpora (Heid et al., 2004). Furthermore, it allows for cross-level queries and accounts for intersecting hierarchical annotations.

NXTSearch thus offers many of the functionalities that ANNIS aims to supply. Nevertheless it does neither provide the means for visualizing and querying the annotation in a user-friendly way, nor is it accessible via the internet.

ANNIS aims at combining the advantages of the presented systems. As a web-based interface, it provides easy and quick access to linguistic data via the internet. Future development of ANNIS will profit from experiences in the user-friendly design of tools such as TIGERSearch, eventually arriving at a tool that can be easily used by non-experts. Similarly, ANNIS will build upon and continue work on multi-level and cross-level querying of tools such as NXTSearch.

7 Summary and Future Directions

We have characterized the application scenario for the ANNIS linguistic database, explained the ensuing design decisions, and described the present state of the implementation. This first version is now ready for use within the SFB and will be further developed in accordance with users' experiences. Specifically, we plan to undertake usability studies regarding both the query facilities and the visualization scheme used in the present implementation. We expect that these two topics are the central ones for further improving the system.

[30] `http://www.ims.uni-stuttgart.de/projekte/nite/manual/`

For querying, an option to consider is providing two different ways of accessing data: a formal query language that allows experienced users to quickly construct the expression they are interested in, and a more user-friendly one for inexperienced users, which might offer graphical options (like in TIGERSearch) and interactive help facilities. The two user groups have very different requirements, so that providing tailored access languages seems appropriate.

As for visualization, a better way of displaying trees should be integrated. Similarly, provisions have to be made to display discourse-related annotations more effectively. Co-reference information, for instance, could be shown by colouring the co-referring expressions in the discourse view (as in the MMAX annotation tool).

Within the SFB, various working groups are developing standardized tag sets and annotation guidelines (as discussed in Section 2.2). Step by step, these will be integrated into ANNIS, with the annotation guidelines made available so that users can interpret annotations that are not their own.

At least in the first round of data annotation, it might become necessary to modify the SFB questionnaire or annotation guidelines and adapt them to unforeseen data. ANNIS should thus provide a suitable way of handling data that is annotated according to different versions of the questionnaire or guidelines.

Also, some further kinds of data have to be integrated into the database:

- The questionnaire mentioned in Section 2.2 should be mapped to ANNIS so that answers can be looked for in the context of their questions; also, the hierarchical structure of the questionnaire should be preserved.

- Speech data at the moment is ‘integrated’ only by a hyperlink to a sound file, which might not be sufficient in the long term.

- When data in many languages is added to ANNIS, it becomes relevant to add typological information, which could then be used in the queries.

On the technical side, an important step will be adding a database to the system for application scenario A (with a centralized data repository), to ensure that ANNIS be ready to hold larger amounts of data than is possible in the present RAM-based version. Furthermore, *metadata* has to be systematically integrated into the data structures, possibly with ramifications for the query language (e.g., provide the ability to search data that originated before a specific date). Once again, existing standards such as TEI, IMDI[31], OLAC[32] will inform the design decisions.

Bibliography

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, Sozopol, 2002.

Stefanie Dipper, Michael Götze, and Manfred Stede. Simple annotation tools for complex annotation tasks: an evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, pages 54–62, Lisbon, 2004.

Michael Götze. Zur Annotation von Informationsstruktur, 2003. Diploma thesis, Universität Potsdam, Institut für Linguistik.

Juliane Gross. Algorithmen zur Behandlung von Anaphora in Zeitungskommentaren, 2003. Diploma thesis, Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik.

Ulrich Heid, Holger Voormann, Jan-Torsten Milde, Ulrike Gut, Katrin Erk, and Sebastian Padó. Querying both time-aligned and hierarchical corpora with NXT Search. In *Proceedings of the Forth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1455–1458, Lisbon, 2004.

[31] `http://www.mpi.nl/IMDI/`
[32] `http://www.language-archives.org/`

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

Manfred Stede. The Potsdam Commentary Corpus. In *Proceedings of the ACL Workshop on Discourse Annotation*, pages 96–102, Barcelona, 2004.

Manfred Stede and Silvan Heintze. Machine-assisted rhetorical structure annotation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 425–431, Geneva, 2004.

Ilona Steiner and Laura Kallmeyer. VIQTORYA—a visual query tool for syntactically annotated corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1704–1711, Las Palmas, 2002.

Stefanie Dipper, Michael Götze and Manfred Stede
Universität Potsdam
SFB 632, Institut für Linguistik
Postfach 601553
14415 Potsdam
Germany
{dipper,goetze,stede}@ling.uni-potsdam.de

Tillmann Wegst
Max-Braun-Str. 36
D-66123 Saarbrücken
Germany
post@tillmann-wegst.de