Preface

The papers in this volume were presented at the workshop *Heterogeneity in Linguistic Databases*, which took place on July 9, 2004 at Potsdam University. The workshop was organized by project D1: *Linguistic Database for Information Structure: Annotation and Retrieval*, a member project of the SFB 632, a collaborative research center entitled *Information Structure: the Linguistic Means for Structuring Utterances, Sentences and Texts*.[1]

A prominent feature of the SFB 632 is that it unites projects that base their research on empirical data. The individual projects collect and annotate linguistic data of various types, which then constitutes the ground for further research. A central project provides the technical infrastructure for building, maintaining, and retrieving this data. Similar to many current comparable research projects, the SFB deals with very heterogeneous linguistic data. This heterogeneity results from a number of factors: First, primary data itself is heterogeneous, differing with respect to size (e.g., single sentences vs. sentences in context), modality (monologue vs. dialogue, text vs. speech), and language. Second, rich, multi-level annotations often require data structures of various types (attribute-value pairs, trees, pointers, etc.). Third, data is often annotated by means of different, task-specific annotation tools (e.g., by tools for syntax, discourse, or co-reference annotation). Furthermore, similar to the data, the needs and backgrounds of potential users of the data vary: Researchers come with tasks of diverse types—ranging from manual exploration to automatic statistical computations—and differ with regard to their computer skills. This diversity must be reflected by the retrieval facilities.

The workshop brought together developers and users of linguistic databases from a number of research projects that work on an empirical basis, all of which have to cope with some of the issues sketched out above. The first four papers address aspects of heterogeneous data from the point of view of database developers; the remaining three papers focus on data exploitation by the users.

In his paper *Unity in Diversity: Integrating Differing Linguistic Data in TUSNELDA*, Andreas Wagner presents the TUSNELDA corpus, a collection of diverse sub-corpora, which differ with regard to object languages, text types,

[1] For more information about the SFB 632, visit `http://www.sfb632.uni-potsdam.de/`, for project D1, see `http://www.sfb632.uni-potsdam.de/projects/d1`.

annotation types, and underlying linguistic theories. The underlying, XML-based annotation scheme TUSNELDA is both sufficiently flexible to cover the diversity of the data as well as uniform enough to enhance data (re)usability.

Thomas Schmidt's paper *EXMARaLDA und Datenbank ‚Mehrsprachigkeit' — Konzepte und praktische Erfahrungen* (in German, with extended abstract in English) addresses concepts and principles in the development of a database for heterogeneous speech data, such as the importance of standardization at different levels of data processing. He also reports practical experiences with users and technologies, including time costs in developing the database.

In her paper *Heterogeneity and Standardization in Data, Use, and Annotation: a Diachronic Corpus of German*, Anke Lüdeling describes problems specific to diachronic corpora, such as the highly variable orthography of old texts. She proposes a flexible corpus design that encompasses this variability and, at the same time, supports standardized annotation.

Multiple Hierarchies: New Aspects of an Old Solution by Andreas Witt addresses problems posed by data annotations with (i) multiple/overlapping hierarchies and (ii) heterogeneous tagsets. He discusses different SGML/XML-based solutions and proposes a redundant encoding, which replicates the primary data for each hierarchy and tagset and makes use of Prolog facts for data representation.

Roland Meyer's paper *VP-Fronting in Czech and Polish—A Case Study in Corpus-Oriented Grammar Research* shows how corpus data can be used for the examination of a specific linguistic phenomenon. He examines VP-fronting in Czech and Polish and its information-structural facets. Referring to different corpora with varying types of annotation, he discusses user requirements with regard to annotations and search facilities.

George Smith (*Refining Queries on a Treebank with XSLT Filters. Approaching the Universal Quantifier*) argues for the importance of the universal quantifier in query languages, which is a prerequisite for (easy) retrieval of many linguistic phenomena but is often not implemented due to efficiency reasons. He presents XSLT stylesheets that implement the universal quantifier; these stylesheets can be used, for example, to further filter the results of a TIGERSearch query.

In their paper *Exploring Lexical Patterns in Text: Lexical Cohesion Analysis with WordNet*, Elke Teich and Peter Fankhauser present an analysis and exploration of lexical cohesion. First, lexical chains are annotated in a corpus, based on WordNet. A specific tool then allows the user to analyze the chains manually. Statistic analyses of the data show, among other things, that length and type of lexical chains depend on the register of the text.

We are very grateful to the authors for their contributions to this volume. We would also like to thank them for their presentations at the workshop, which stimulated some very interesting and fruitful discussions.

Stefanie Dipper
Michael Götze
Manfred Stede

Contents

Unity in Diversity: Integrating Differing Linguistic Data in TUSNELDA

Andreas Wagner

Universität Tübingen

This paper describes the creation and preparation of TUSNELDA, a collection of corpus data built for linguistic research. This collection contains a number of linguistically annotated corpora which differ in various aspects such as language, text sorts / data types, encoded annotation levels, and linguistic theories underlying the annotation. The paper focuses on this variation on the one hand and the way how these heterogeneous data are integrated into one resource on the other hand.

1 Introduction

The principal concern of the collaborative research centre (Sonderforschungsbereich) SFB 441 at University of Tübingen are the empiric data structures which feed into linguistic theory building. In order to approach this general issue from a considerable variety of research perspectives, SFB 441 comprises different projects each of which empirically investigates a particular linguistic phenomenon in a particular language or language family. The respective research interests range from suboptimal syntactic structures in German, local and temporal deictic expressions in Bosnian/Croatian/Serbian or Portuguese and Spanish, to semantic roles, case relations, and cross-clausal references in Tibetan, to mention just a few. As empirical basis for their research, many projects create electronically accessible collections of linguistic data and prepare them to fit their particular needs. In most cases, these collections are corpora. However, a couple of projects deal with data (e.g. lexical information) which are more adequately represented by an Entity-Relationship based data model and thus are implemented in relational databases rather than corpora.

Interdisciplinary Studies on Information Structure 02 (2005): 1–20

Dipper, S., M. Götze and M. Stede (eds.):

Heterogeneity in Focus: Creating and Using Linguistic Databases

All data collections built within SFB 441 projects are assembled in one repository called TUSNELDA (= *TUebinger Sammlung Nutzbarer Empirischer Linguistischer DAtenstrukturen*, Tübingen collection of reusable, empirical, linguistic data structures). Especially, the different corpora are integrated into a common XML-based environment of encoding, storage, and retrieval. This integration is particularly challenging due to the heterogeneity of the individual corpora, which differ with regard to the following aspects:

- languages (e.g. German, Russian, Portuguese, Tibetan,...)
- text types / data types (e.g. newspaper texts, diachronic texts, dialogues, treebanks, ...)
- categories of information covered by the annotation / annotation levels (e.g. layout, textual structure, morpho-syntax, syntax, ...)
- underlying linguistic theories

This paper describes the approach pursued to integrate these heterogeneous corpus data. Section 2 provides an overview of the corpora built by the individual projects. This overview illustrates the diversity of the data. Section 3 addresses their integration in TUSNELDA. In particular, aspects of the annotation process, the annotation schemes and the underlying data model, as well as corpus management and retrieval are discussed.

2 SFB 441 Corpora

This section provides an overview of the different corpora created in SFB 441. In the following listing, each project engaged in corpus building is mentioned together with the investigated language and the respective corpora. For each corpus, a short general description is given, including its size and a list of the annotation levels encoded in it.

Project A1: “Representation and automatic acquisition of linguistic data”
German

- **TüBa-D/Z (Tübinger Baumbank des Deutschen / Zeitungstexte)**
  manually annotated treebank (approx. 15,000 sentences)
  - syntactic structures

- **TüPP-D/Z (Tübinger Partiell Geparstes Korpus des Deutschen / Zeitungstexte**
  newspaper corpus; syntactically analysed by means of a rule-based chunk parser created in the project (approx. 200 million words; only partially integrated in TUSNELDA)
  - text structures (paragraphs, sentence boundaries, etc.)
  - syntactic structures

Project A3: “Suboptimal syntactic structures”
German

- **Database of Grammaticality Judgements**
  manually annotated example sentences originating from linguistic literature including grammaticality judgements (approx. 1,000 sentences)
  - morphological features (e.g. case)
  - syntactic structures

Project B1: “Corpus based study of address and linguistic politeness in the Slavonic languages”
Russian

- **Russian Interviews**
  interviews in newspapers (approx. 290,000 words)
  - text structures

- **Uppsala Corpus of Modern Russian**
  balanced Russian corpus compiled in Uppsala; extended by morpho-syntactic annotation by means of a POS tagger created in the project (1 million words)
  - text structures
  - morphological features / POS tags

Project B3: "Modal verbs and modality in German"
German

- **Goetz von Berlichingen**
  Early New High German text, digitised for the TITUS project (approx. 43,000 words)
  - text structure
  - layout (page and line breaks)

Project B8: "Corpus-based analysis of local and temporal deictics in (spontaneously) spoken and (reflected) written language"
Bosnian/Croatian/Serbian

- **Tübinger BKS-Korpus**
  Comic Corpus, Bosnian Interviews, Novosadski korpus of Spoken Language (approx. 127,000 words)
  - text structure / dialogue structure
  - marking and classification of deictic expressions
  - situational context (accompanying gesture)

Project B9: "Local and temporal deixis in the Romance languages — History and variation"
Portuguese, Spanish

- **TüPoDia (Tübinger Portugiesische Diachrone Texte)**
  Portuguese diachronic texts (approx. 260,000 words)
  - text structure
  - marking and classification of deictic expressions

- **BraToLi (Brasilien Toledo Lima)**
  transcriptions of spoken dialogs (including situational descriptions) from Brasil, Toledo, and Lima (approx. 10,000 words)
  - dialogue structure
  - marking and classification of deictic expressions
  - situational context

Project B11: "Semantic roles, case relations, and cross-clausal reference in Tibetan"
Tibetan

- **Tibetan Corpus**
  texts from different regions and epochs (currently approx. 700 clauses, to be extended)
  - text structure
  - layout (page breaks)
  - morphological features (e.g. case)
  - syntactic structures
  - verb–argument structures
  - cross-clausal references (anaphoric reference via empty arguments and pronouns)

For some of these corpora, substantial extensions are envisaged to cover additional annotation levels. For example, the German treebank TüBa-D/Z will be extended by co-reference annotation; the Tibetan corpus will be augmented by

lexical resources and English translations, which will be aligned to the annotated texts.

3 Integration in TUSNELDA

All the corpora mentioned in the previous section form the components (sub-corpora) of the TUSNELDA corpus. This means that they are integrated into a common environment regarding annotation, data management, and corpus querying. This environment is based on XML technology. This has two major advantages. Firstly, XML offers the flexibility required to encode all the peculiarities of the heterogeneous data sketched above. Secondly, various software for encoding, managing and querying XML documents is available and can be employed. The alternative, developing and implementing such software from scratch, appears infeasible in view of the diversity of requirements for encoding and processing the different corpora.

In detail, the integration of the different corpora involves several stages:

1. development of unified annotation schemes which cover all (combinations of) annotation levels realised in the TUSNELDA sub-corpora
2. transformation of the individual corpora into a format which obeys the respective annotation schemes
3. storing and managing the TUSNELDA sub-corpora in an XML database
4. implementation of query interfaces which are tailored to the respective annotation levels to be searched

3.1 Annotation Process

As noted in section 1, the individual sub-corpora of TUSNELDA are built separately in the respective SFB 441 projects. Moreover, their diversity implies

that different annotation procedures are most adequate and efficient in the respective corpus building activities. In this respect, two basic scenarios can be distinguished:

In one scenario, a proprietary data format and corresponding proprietary software is employed for annotation. This is appropriate in case there is an established way of annotating the information to be covered by the corpus, and in case a common and convenient annotation tool is available which supports this annotation. For example, the projects that create syntactic treebanks employ *annotate* (cf. Plaehn (1998)) for that task. This tool is widely used for building collections of syntactic trees. It provides a number of convenient features which speed up annotation, such as a graphical interface and facilities for interactive semi-automatic annotation. *annotate* encodes the data in the proprietary NEGRA format. A special case of this scenario is the use of tools for automatic annotation, such as POS taggers or shallow parsers, which of course require specific input and output formats. Integrating corpora built that way in TUSNELDA comprises two steps. Firstly, annotation schemes have to be developed and/or adapted to cover all information encoded in the corpora. Secondly, the corpora have to be converted from their respective proprietary format into XML structures which are conforming to the corresponding TUSNELDA annotation scheme. As a general rule, this format conversion can be done automatically.

In the alternative scenario, annotation immediately rests upon the TUSNELDA annotation schemes, i.e. TUSNELDA-conforming XML markup is created directly. This procedure is appropriate if a common practice for annotating the sort of information to be encoded in the corpus does not yet exist. Guided by their specific research interests, some projects create corpora which cover certain peculiar aspects (or combinations of aspects) for which neither an established annotation scheme nor a tailored annotation tool is available. For example, it was all but clear in advance how to adequately encode the closely interrelated aspects of syntactic structure, verb–argument structure and cross-

clausal reference in the Tibetan corpus. To handle such cases, a preliminary annotation scheme is developed in advance (as a DTD), and annotation is performed according to this scheme, using a general XML editor. In the course of the annotation process, with growing experience regarding the data, it usually turns out that revisions and extensions of the provisional scheme are necessary to appropriately encode certain peculiarities and/or to improve the possibilities of retrieving interesting information. Thus, the scheme is incrementally adapted to these emerging requirements. In this scenario, the annotation generally has to be performed manually. However, to increase efficiency, we aim at automatising annotation steps wherever possible (e.g. assigning unique IDs to elements). As annotation software we mainly use the CLaRK system (cf. Simov et al. (2001)), an XML editor which has been developed especially for encoding linguistic resources. On the one hand, this tool is not restricted to specific formats but supports any XML DTD. On the other hand, it comprises a number of facilities to perform annotation steps automatically or semi-automatically, such as regular grammar engines or constraint mechanisms which add specific markup depending on the context. These facilities are flexibly configurable and adaptable to the particular annotation scheme in use.[1]

3.2 TUSNELDA Annotation Scheme

Various general requirements guide the definition of annotation schemes for TUSNELDA. First of all, these schemes have to be exhaustive, i.e. they must capture all kinds of information which is encoded within the different annotation levels in the TUSNELDA corpora. As a second crucial requirement, the schemes should be convenient with respect to both annotation and retrieval. This means they should be designed in a way which facilitates manual anno-

[1] Wagner and Zeisler (2004) outline how these facilities are employed for annotating the Tibetan Corpus.

tation and allows the specification of “intuitive” search queries. These criteria imply two further requirements, which in a sense are complementary to each other. On the one hand, the schemes have to be open for different languages and linguistic theories. This is necessary since TUSNELDA is multilingual and its corpora are based upon differing theoretic approaches. On the other hand, analogous structures and phenomena in the different corpora should be encoded in analogous ways. This enhances reusability because it allows for the development of common mechanisms for annotation or format conversion as well as the implementation of analogous retrieval interfaces (including the specification of similar—if not identical—search queries) for the different corpora. In addition, keeping the annotation schemes as uniform as possible reveals commonalities and deviations of the information encoded in the different corpora.

Despite the diversity of the corpora in TUSNELDA, they all share the same generic data model: hierarchical structures. It is most appropriate to encode the phenomena captured in the TUSNELDA corpora by means of nested hierarchies, augmented by occasional “secondary relations” between arbitrary nodes in these hierarchies. This distinguishes TUSNELDA fundamentally from corpora whose annotation is based on other data models such as, for example, timeline-based markup of speech corpora or multimodal corpora (e.g. cf. Schmidt (2004)). Such corpora encode the exact temporal correspondence between events on parallel layers (e.g. the coincidence of events in speech and accompanying gesture or the overlap of utterances) whereas hierarchical aspects are secondary. In TUSNELDA, however, hierarchical information (e.g. textual or syntactic structures) is prevalent, while capturing the exact temporal coincidence of different events in general is not of primary relevance in the research within SFB 441.

Guided by these requirements, we decided to develop annotation schemes which encode information as embedded annotation (i.e. the markup is placed locally at or around the corresponding text) rather than standoff annotation (where

the markup is stored in a separate file, including pointers to the primary text). Essentially, this decision rests on two major considerations.

The first consideration concerns the required suitability of the schemes for manual annotation in particular and corpus processing in general. While standoff annotation appears to become a "quasi standard" paradigm for linguistic annotation, there is still a lack of general software supporting this paradigm. Usually, projects engaged in standoff annotation develop their own software which is tailored to their specific needs. Such software would, if at all, be only of limited use for annotating a corpus in TUSNELDA. Furthermore, due to the diversity of our corpora, we need general XML-aware tools which are adaptable to particular requirements of each individual corpus. Currently, such tools (XML editors, format conversion tools, XML databases and query engines) are optimised for processing hierarchical XML structures, i.e. they are well suited for embedded annotation, while providing at best rudimentary support for standoff annotation.

The second consideration is the fact that embedded annotation indeed is sufficient for encoding our data. Standoff annotation would be necessary if the structures to be encoded formed overlapping hierarchies, which cannot be modelled within a single XML document. Actually, this problem does not arise for our data. The structures primarily encoded in the TUSNELDA corpora are at the textual and/or syntactic level. Since syntactic structures constitute subsentential hierarchies while text structures define super-sentential hierarchies, these structures do not overlap so that they can be captured within a single document hierarchy. Concurrent hierarchical units occur only marginally and are not of primary importance. These units concern the physical (layout) structure of the annotated texts, e.g. page boundaries. Such boundaries are marked by empty XML elements (e.g. <pb/> for a page break), which do not violate the well-formedness of the document.

```
<s>
  <clause>
    <ntNode>
      <tok>
        <orth>khra·phru·gu</orth>
        <pos>NOM:anim∼pers</pos>
      </tok>
      <ntNodeCat>NP</ntNodeCat>
      <desc>
        <case>Abs</case>
      </desc>
    </ntNode>
    <tok id="v6">
      <orth n="2">med-tshug</orth>
      <pos>VFIN</pos>
      <desc>
        <feature type="part">NEG</feature>
        ...
      </desc>
    </tok>
    <clauseCat>simple</clauseCat>
  </clause>
  <punct>|</punct>
</s>
```

Figure 1: Example annotation from Tibetan corpus (1)

3.3 Examples

This section provides several examples which illustrate diverse (combinations of) annotation levels captured in the individual corpora and how these different sorts of information are encoded. These examples will also illustrate how the balance between the desired uniformity and the required flexibility w.r.t. different languages and theories is achieved.

Figure 1, taken from the Tibetan Corpus, exemplifies the encoding of syntactic structures. <tok> elements mark the tokens (i.e. words) of a text with their orthographic or phonemic realisation (<orth>) and part-of-speech classification (<pos>). A phrase is encoded by an <ntNode> (non-terminal node) element; <ntNodeCat> marks its category. For clausal constituents, there is a special element <clause> (including <clauseCat> specifying the clause category).[2] <ntNode> and <clause> elements may be recursively nested. Tokens, phrases, and clauses may receive a further linguistic description (<desc>). Such descriptions may contain simple features like case[3] or complex specifications like the argument structure of a verb.

An example for the encoding of argument structures in the Tibetan Corpus is shown in figure 2. This encoding belongs to the annotation displayed in figure 1. In fact it is located within the <desc> element of the verb token (at the position indicated by the dots) and presented here in a separate figure just for the sake of clarity. (This exemplifies the integration of different annotation levels—syntactic constituent structures and verb–argument structures—in one XML hierarchy.) In detail, the description comprises (a) the "canonical" argument structure (a list of <complement> elements within a <frame> element), and (b) the "real" frame, i.e. the realisation of the arguments in the clause, including additional arguments (a list of <realComplement> elements within a <realFrame> element). Each <complement> element within <frame> has a corresponding <realComplement> element within <realFrame> (possibly marked as not realised in the respective clause, see below). The order of <realComplement> items corresponds to the order of the respective <complement> items; additional complements which occur in the clause but are not included in the canon-

[2] In some corpora, no explicit distinction is made between clausal and other constituents; in these corpora, clauses are annotated as <ntNode> instead of <clause>.

[3] A certain set of common features is defined in the annotation scheme by specific elements such as <case>, <number>, or <person>. Furthermore, a general element <feature> (with a 'type' attribute) allows the specification of any kind of feature.

```
...
<frame>
   <complement>
      <role>POSS</role>
      <case>Aes</case>
   </complement>
   <complement>
      <role>EXST2</role>
      <case>Abs</case>
   </complement>
</frame>
<realFrame>
   <realComplement id="v6c1" status="empty">
      <role>POSS</role>
      <ref target="v5c1"> </ref>
   </realComplement>
   <realComplement id="v6c2">
      <role>EXST2</role>
   </realComplement>
</realFrame>
...
```

Figure 2: Example annotation from Tibetan corpus (2)

ical frame are represented by <realComplement> elements appended at the end of the <realFrame> list. In case the order of complements as realised in the clause deviates from the canonical complement order as defined in <frame>, <realFrame> receives the attribute 'order', which encodes the complement order in the clause (as a sequence of role labels).

For each canonical and real complement, the semantic role is specified. Furthermore, each canonical complement receives a specification of its case. The encoding of argument structure also captures information about cross-clausal references, especially the relation between empty arguments (i.e. arguments not

<figure id="*s45b3*">
  <figTrans>
    <sp who="*Komandant*">
      <spokenPar>
        Nadam se da govoriš istinu . . . Idite , potražite taoca , a <marked type="*deic-dem*">**ovu**</marked> **dvojicu u zatvor !**
      </spokenPar>
      <situation>
        <keywords>
          <term>*open hand*</term>
          <term>*stretched out*</term>
        </keywords>
      </situation>
    </sp>
  </figTrans>
</figure>

Figure 3: Example annotation from BKS Korpus (Comic Corpus)

overtly realised in a clause) and their antecedents in previous clauses.[4] To capture this kind of cross-clausal reference, each <realComplement> receives a unique ID. Empty arguments (e.g. the first <realComplement> in the example) receive an attribute marking emptiness and a pointer to the corresponding antecedent in the text, which in most cases is a <realComplement> specified in the argument structure of some previous clause. Such a pointer is encoded as a reference tag (<ref>) with an attribute 'target' that points to the ID number of the corresponding referee.

Figure 3 displays the encoding of a single comic picture in the BKS Comic Corpus. This encoding significantly differs from the previous examples in the

[4] The investigation of this phenomenon is one of the major research interests of project B11, which is building the Tibetan Corpus.

covered annotation levels; instead of entirely capturing complex syntactic structures, it provides punctual information about specific expressions (in this case deictics) and the situational context of their usage, especially accompanying gesture. A comic picture (captured by a <figure> element) is represented by a transcription (<figTrans>) of the dialogue taking place in this picture.[5] Each dialogue turn is encoded by a <sp> element with an attribute 'who' indicating the speaker. The utterance is captured by a <spokenPar> (spoken paragraph) element. Expressions of specific interest, as deictic expressions in the BKS Corpus, can be marked by the element <marked>; the attribute 'type' provides a classification of the expression. In the example, the word "ovu" is marked as demonstrative deictic ("*deic-dem*"). The element <situation> contains information about the situational context. In the Comic Corpus, this information is encoded as a set of keywords (a list of <term> elements within a <keywords> element) specifying gesture accompanying deictics. Note that this kind of transcription basically makes use of a hierarchical scheme rather than a timeline-based scheme employed for other transcriptions of dialogue. The research purpose which guided the creation of the Comic Corpus, i.e. the examination of deictic expressions and co-occurring pointing gesture, does not require the encoding of exact temporal overlaps between different utterances and/or nonverbal events. For this reason, the transcription of comics, where such temporal overlap is not determinable, is suitable for the research intended.

Figures 4 and 5 illustrate the openness of the TUSNELDA annotation schemes for different linguistic theories. Each of these figures shows a syntactic tree of a sentence: figure 4 from the TüBa-D/Z treebank, figure 5 from the Database of Grammaticality Judgements. Both sentences are in German and have considerable commonalities (wh-element "wie", "dass"-clause with

[5] More exactly, this transcription includes all written material, i.e. spoken utterances as well as text displayed on some artefact, e.g. a board, and "meta-situational" comments of the author located on top or bottom of the picture.

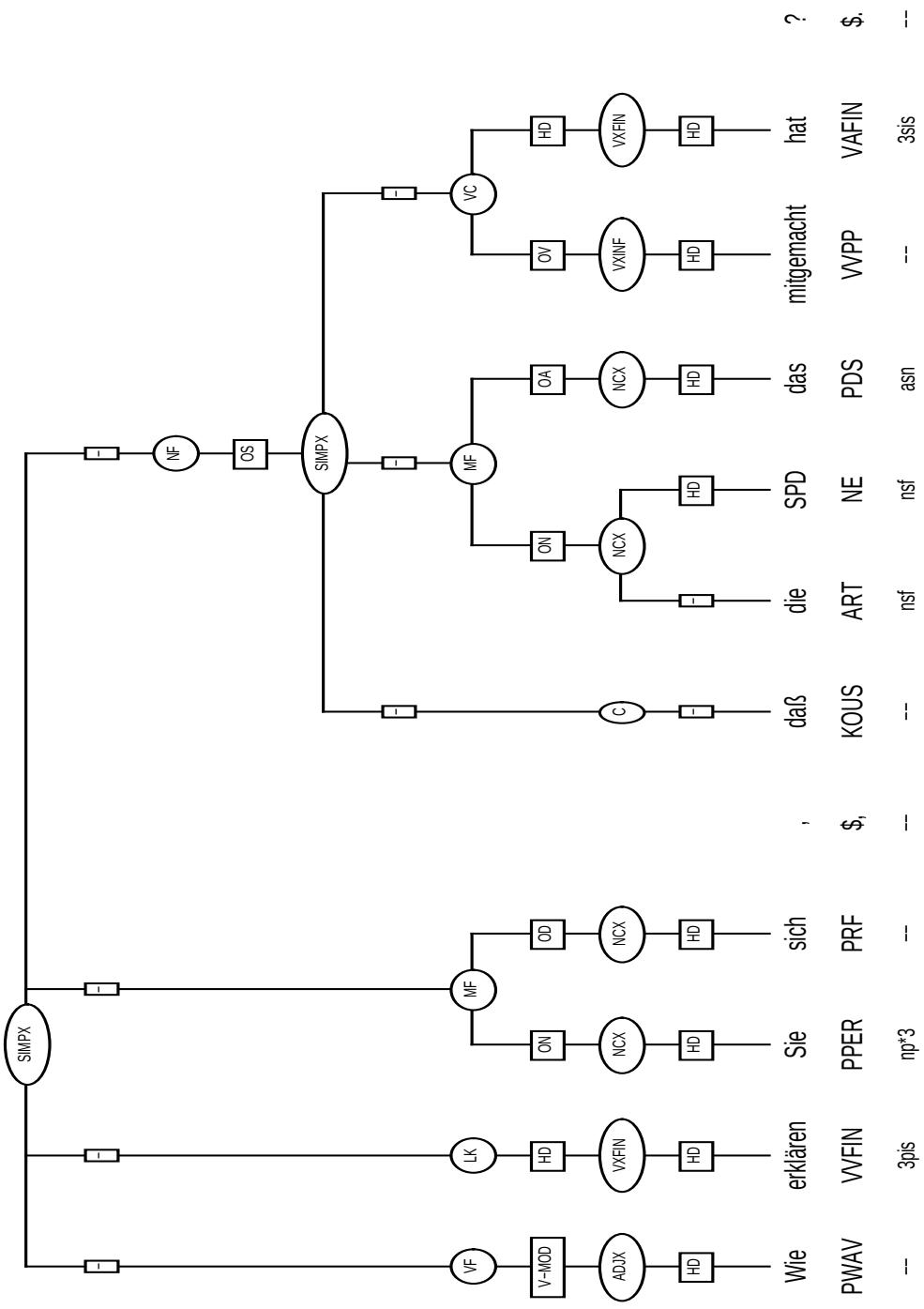


Figure 4: Example tree from TüBa-D/Z

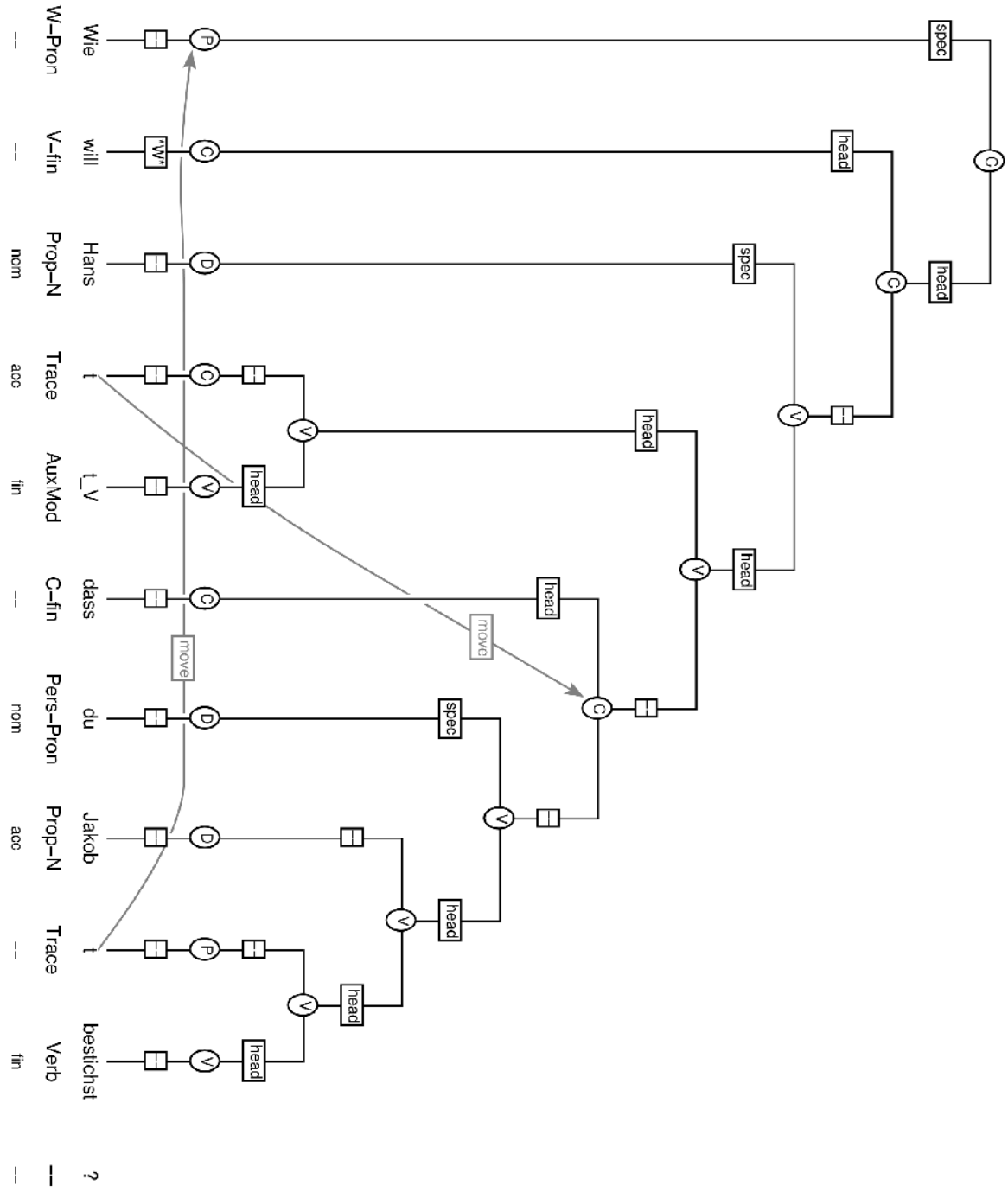


Figure 5: Example tree from the Database of Grammaticality Judgements

transitive verb). However, they are assigned very different syntactic structures, which reflect the linguistic theories and assumptions underlying the two treebanks. The annotation in TüBa-D/Z is guided by the theory of topological fields (a traditional descriptive theory accounting for the constituent order in German sentences) and the restriction to context-free structures, which results in comparably flat structures without traces. In contrast, the Database of Grammaticality Judgements is intended to comprise trees in accordance with generative syntax, characterised by highly nested (usually binary-branched) structures and the common use of traces. The TUSNELDA annotation scheme for syntactic structures is compatible to both approaches, i.e. both trees can be represented by an XML structure as in figure 1. The TUSNELDA scheme neither prescribes a set of POS tags and constituent labels nor constrains the configuration of syntactic trees. The only restrictions it imposes on the encoding of syntactic structures is the distinction between tokens (words) and non-terminal nodes (with the additional possibility to identify clause nodes by a special element) and the limitation to tree structures with possible secondary edges. These constraints mark the balance between the desirable uniformity and the required flexibility which is appropriate for TUSNELDA and its corpora.

3.4 Corpus Management and Querying

After the step of annotation (and, if necessary, format conversion), a corpus can be imported into a database which serves as the central platform for managing and querying the TUSNELDA corpora. As database software we employ *Tamino XML Server* developed by Software AG. Tamino is a native XML database and implements several techniques for indexing XML documents. This allows an efficient search in the data. Furthermore, Tamino provides a query language which is a subset of XQuery (cf. Boag et al. (in progress)). XQuery is being developed to serve as the standard language for querying XML data.

As Sasaki et al. (2004) point out, XQuery is particularly suited for retrieving hierarchical aspects of annotated material, which renders it less useful for corpora which are not based upon hierarchical data models. However, as discussed above, the annotation in TUSNELDA essentially is hierarchically organised so that XQuery is an appropriate query language.

The data in the TUSNELDA collection are made publicly accessible via a WWW interface (www.sfb441.uni-tuebingen.de/tusnelda.html). The Tamino software offers various facilities to configure HTTP-based interfaces for searching the XML database and formatting the query results. We employ these facilities to realise web interfaces which take into account the respective peculiarities of the individual corpora. The core of the search mechanism is the XQuery engine of the database. The user can formulate queries in a format based on XPath and XQuery. Concerning general accessibility of the interface, it makes more sense to rely on these standard languages for querying XML data than on proprietary query languages. However, the prospective users of TUSNELDA, i.e. linguistic and philological researchers, are usually not familiar with these languages. Therefore, we extend the interface with various mechanisms which render the interface more user-friendly. For instance, we provide corpus-specific example queries as well as templates and syntactic abbreviations which facilitate the formulation of "typical" queries. Furthermore, the user can choose between alternative formats of output display (e.g. syntactic structures can be viewed as graphical trees, labelled bracket structures, or XML structures as annotated in the corpus). Such facilities and their suitability to improve user-friendliness will be subject to the feedback by actual and prospective users inside and outside SFB 441. In this sense, the current WWW interface is in a preliminary state and will continually be refined to improve its benefit for the linguistic research community.

Bibliography

Scott Boag, Don Chamberlin, Mary Fernández, et al. XQuery 1.0: An XML Query Language. W3C working draft. Technical report, W3C, in progress. URL `http://www.w3.org/TR/xquery/`.

Oliver Plaehn. *Annotate Bedienungsanleitung*. Universität des Saarlandes, Sonderforschungsbereich 378, Projekt C3, Saarbrücken, Germany, April 1998.

Felix Sasaki, Andreas Witt, Dafydd Gibbon, and Thorsten Trippel. Concept-based queries: Combining and reusing linguistic corpus formats and query languages. In *Proc. of LREC 2004*, pages 259–262, Lisboa, May 2004.

Thomas Schmidt. Transcribing and annotating spoken language with EXMARaLDA. In *Proc. of LREC 2004 Workshop on XML-based Richly Annotated Corpora*, pages 69–74, Lisboa, May 2004.

Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, and Atanas Kiryakov. CLaRK - an XML-based system for corpora development. In *Proc. of the Corpus Linguistics 2001 Conference*, pages 558–560, 2001.

Andreas Wagner and Bettina Zeisler. A syntactically annotated corpus of Tibetan. In *Proc. of LREC 2004*, pages 1141–1144, Lisboa, May 2004.

Andreas Wagner
Universität Tübingen
SFB 441
Nauklerstr. 35
72074 Tübingen
Germany
wagner@sfs.uni-tuebingen.de
http://www.sfb441.uni-tuebingen.de/~wagner

EXMARaLDA und Datenbank ‚Mehrsprachigkeit' – Konzepte und praktische Erfahrungen*

Thomas Schmidt

SFB 538 ‚Mehrsprachigkeit', Universität Hamburg

This paper presents some concepts and principles used in the development of a database of multilingual spoken discourse at the University of Hamburg. The emphasis of the first part is on general considerations for the handling of heterogeneous data sets: After showing that diversity in transcription data is partly conceptually and partly technologically motivated, it is argued that the processing of transcription corpora should be approached via a three-level architecture which separates form (application) and content (data) on the one hand, and logical and physical data structures on the other hand. Such an architecture does not only pave the way for modern text-technological approaches to linguistic data processing, it can also help to decide where and how a standardization in the work with heterogeneous data is possible and desirable and where it would run counter to the needs of the research community. It is further argued that, in order to ensure user acceptance, new solutions developed in this approach must take care not to abandon established concepts too quickly.

The focus of the second part is on some practical experiences with users and technologies gained in the four years' project work. Concerning the practical development work, the value of open standards like XML and Unicode is emphasized and some limitations of the "platform-independent" JAVA technology are indicated. With respect to users of the EXMARaLDA system, a predominantly conservative attitude towards technological innovations in transcription corpus work can be stated: individual users tend to stick to known functionalities and are reluctant to adopt themselves to the new possibilities. Furthermore, an active commitment to cooperative corpus work still seems to be the exception rather than the rule.

It is concluded that technological innovations can contribute their share to a progress in the work with heterogeneous linguistic data, but that they will have to be supplemented, in the long run, with an adequate methodological reflection and the creation of an appropriate infrastructure.

* Ich danke den Teilnehmern des Workshops für die fruchtbaren Diskussionen.

Interdisciplinary Studies on Information Structure 02 (2005): 21–42
Dipper, S., M. Götze and M. Stede (eds.):
Heterogeneity in Focus: Creating and Using Linguistic Databases

1 Einleitung

In diesem Aufsatz geht es um die Datenbank ‚Mehrsprachigkeit' und das System EXMARaLDA, die am SFB 538 ‚Mehrsprachigkeit' der Universität Hamburg entwickelt werden. Da deren konzeptuelle und technische Details bereits an anderer Stelle ausführlich dargestellt worden sind (z.B. Schmidt 2004), soll der Schwerpunkt hier einerseits auf solchen Aspekten liegen, die – gemäß dem Thema des Workshops – mit allgemeineren Fragen zum Umgang mit computerverwertbaren, heterogenen linguistischen Datenbeständen zu tun haben. Andererseits soll versucht werden, aus den praktischen Erfahrungen der nunmehr vierjährigen Projektarbeit einige Erkenntnisse abzuleiten, die über den konkreten Projektzusammenhang hinaus für die weitere Arbeit auf diesem Gebiet interessant sein könnten.

2 Daten am SFB ‚Mehrsprachigkeit'

2.1 Überblick

Der Sonderforschungsbereich 538 „Mehrsprachigkeit" vereinigt in seinen vierzehn Teilprojekten eine Vielzahl von Forschern, die sich unter verschiedenen Herangehensweisen dem Thema der Mehrsprachigkeit widmen. In der derzeit laufenden zweiten Förderungsphase (2002-2005) ist der SFB in drei thematische Teilbereiche – „Erwerb der Mehrsprachigkeit", „Mehrsprachige Kommunikation" und „Historische Aspekte der Mehrsprachigkeit" – gegliedert. In ausnahmslos allen Projekten dieser Teilbereiche wird auf empirischer Basis gearbeitet, d.h. Ausgangspunkt der linguistischen Analysen bilden jeweils mehrsprachige Korpora geschriebener oder transkribierter gesprochener Sprache. Die fol-

genden Ausführungen beziehen sich auf die Korpora derjenigen Projekte, die mit gesprochener Sprache arbeiten, genauer:

Projekt	Sprachen	Arbeitsgebiet, theor. Hintergr.	Datentypen (Transk.-System)
K1: Japanische und deutsche Expertendiskurse	Japanisch Deutsch	Diskursanalyse Funktionale Pragmatik	Vortrags- und Planungsdiskurse (HIAT / syncWriter)
K2: Dolmetschen im Krankenhaus	Portugiesisch Türkisch Deutsch	Diskursanalyse Funktionale Pragmatik	Gedolmetschte Arzt-Patienten-Gespräche (HIAT / syncWriter)
K5: Semikommunikation und rezeptive Mehrsprachigkeit im heutigen Skandinavien	Dänisch Schwedisch Norwegisch	Diskursanalyse Funktionale Pragmatik	Radiosendungen, Interviews, Gruppen- und Unterrichtsgespräche (HIAT / HIAT-DOS)
E2: Simultaner und sukzessiver Erwerb von Mehrsprachigkeit	Französisch Portugiesisch Baskisch Spanisch Deutsch	Syntax Generative Grammatik	Spracherwerbsdaten (Interviewer-Kind-Interaktion) (LAPSUS)
E3: Prosodische Beschränkungen zur phonologischen und morphologischen Entwicklung im bilingualen Erstspracherwerb	Spanisch Deutsch	Phonologie Optimalitätstheorie	Spracherwerbsdaten (Interviewer-Kind-Interaktion) (IPA / WordBase)
E4: Spezifische Sprachentwicklungsstörung und früher L2-Erwerb	Deutsch Türkisch	Syntax Generative Grammatik	Spracherwerbsdaten (Interviewer-Kind-Interaktion) (DIGITRAIN / DACODA)
E5: Sprachliche Konnektivität bei bilingual türkisch-deutsch aufwachsenden Kindern	Türkisch Deutsch	Diskursanalyse Funktionale Pragmatik	Spracherwerbsdaten (Evokative Feldexperimente) (HIAT / syncWriter)

2.2 Heterogenität von Transkriptionsdaten

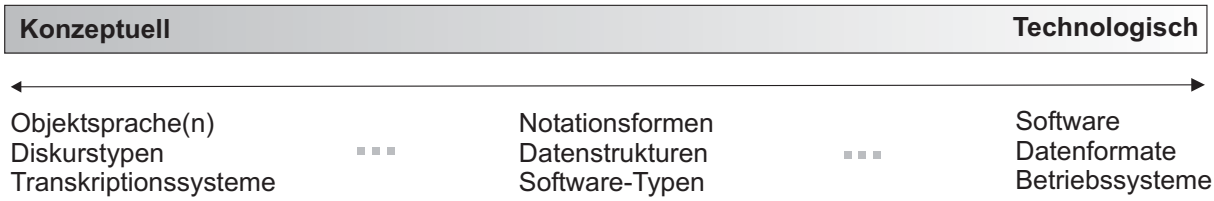
Die obige Tabelle deutet bereits an, dass hinsichtlich der Transkriptionsdaten am SFB eine große Heterogenität besteht. Diese resultiert teils aus *konzeptuellen* Motiven, also aus Unterschieden, die eher (gegenstands-)theoretisch begründet sind, teils aus *technologischen* Motiven, also aus Unterschieden, die eher mit der konkreten technischen Umgebung der praktischen Korpusarbeit zu tun haben.

Ein Beispiel für eine überwiegend konzeptuell bedingte Heterogenität findet sich in den *Transkriptionssystemen*, nach deren Vorgaben Aufnahmen gesprochener Sprache in digitale „Verschriftlichungen" überführt werden. Z.B. benötigen Projekte, die an diskursanalytischen Fragestellungen interessiert sind, für ihre Untersuchungen eine möglichst präzise und nachvollziehbare Repräsen-

tation des zeitlichen Gesprächsablaufs. Sie wählen daher ein Transkriptionssystem, das angemessene Mittel zum Abbilden simultaner Gesprächsbeiträge und non-verbaler Kommunikation beinhaltet – für die diskursanalytisch arbeitenden SFB-Projekte ist dies das mit der Partiturnotation arbeitende System HIAT (Rehbein et al. 2004). Für Projekte, deren Hauptaugenmerk auf phonologischen Fragestellungen liegt, ist hingegen vor allem eine möglichst exakte Repräsentation der lautlichen Gestalt isolierter Redebeiträge wichtig, während der genaue zeitliche Gesprächsablauf in der Transkription modellhaft ausgeblendet werden kann. Das betreffende Projekt E3 arbeitet daher mit einfachen Listen von IPA-transkribierten Äußerungen.

Ausschließlich technologisch bedingt ist hingegen ein Unterschied in der verwendeten *Transkriptionssoftware*: obwohl die Projekte K1, K2 und E5 einerseits und das Projekt K5 andererseits gleiche Arbeitsgebiete und theoretische Hintergründe aufweisen und folglich auch das gleiche Transkriptionssystem (HIAT) verwenden, bewerkstelligten sie die Transkriptionsarbeit ursprünglich mit unterschiedlicher Software (syncWriter bzw. HIAT-DOS); und dieser Unterschied lag ausschließlich in der Rechnerausstattung, genauer: in den jeweils bevorzugten Betriebssystemen (MAC OS bzw. Windows), der Projekte begründet – eine betriebssystemübergreifend einsetzbare Software zum Transkribieren in Partitur-Notation existierte nicht.

Wie die folgende Abbildung illustriert, ordnen sich weitere Aspekte der Heterogenität von Transkriptionsdaten auf einem Spektrum zwischen rein konzeptuell und rein technologisch motivierten Unterschieden ein.



Weiterhin sind diese diversen Aspekte der Heterogenität nicht unabhängig voneinander: Beispielsweise zieht die (rein konzeptuell motivierte) Wahl eines bestimmten Transkriptionssystems oft unweigerlich die (teils konzeptuell, teils technologisch motivierte) Wahl einer bestimmten Notationsform nach sich, die wiederum eine Software erforderlich machen mag, die nicht plattformübergreifend implementiert ist und somit zwangsläufig auch die (eigentlich rein technologisch zu motivierende) Wahl eines Betriebssystems vorbestimmt.

3 Prinzipien und Systemarchitektur der Datenbank ‚Mehrsprachigkeit'

Die Heterogenität der Transkriptionsdaten am SFB erschwert deren Austausch zwischen einzelnen Projekten und stellt somit ein Hindernis für die kooperative Forschung dar: es ist meist nicht ohne Weiteres möglich, die Projektdaten eines Projektes in den Arbeitsumgebungen eines anderen Projektes anzusehen oder auszuwerten, geschweige denn verschiedene Projektkorpora zu vereinen oder mit anderen als den ursprünglich vorgesehenen Werkzeugen zu bearbeiten. Darüber hinaus verhindert die Vielfalt der Formate eine einheitliche und effektive Archivierung der Daten und birgt so die Gefahr, dass aufwändig erstellte Korpora auf lange Sicht unbrauchbar werden.

Ziel des SFB-Projekts ‚Datenbank Mehrsprachigkeit' ist daher die Konzeption und Implementierung einer Plattform für die Erstellung und Auswertung von Korpora gesprochener Sprache, die die älteren projektspezifischen Lösungen ablösen und eine flexible Verarbeitbarkeit, Austauschbarkeit und Archivierbarkeit von Transkriptionsdaten gewährleisten soll.

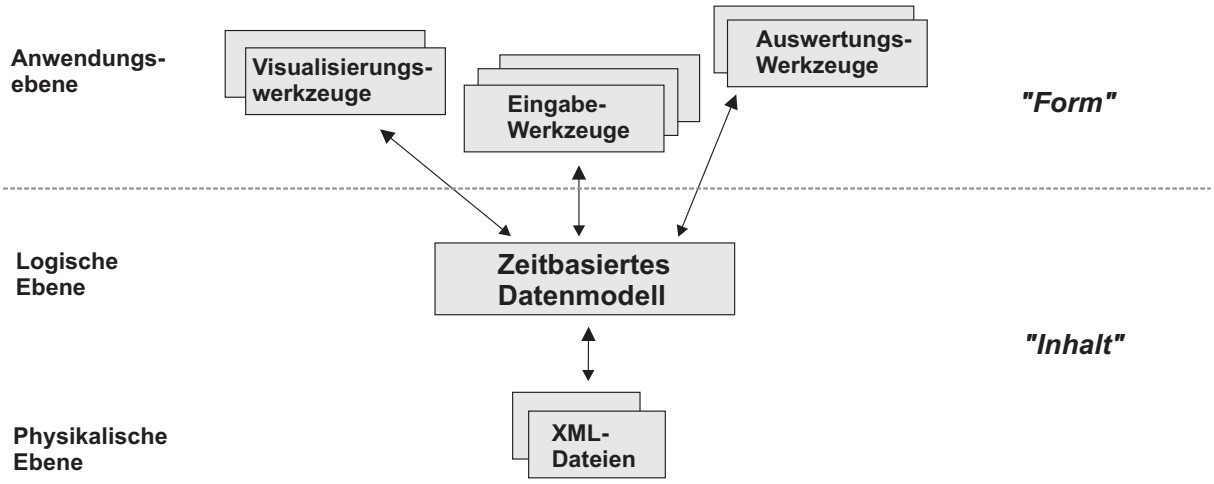
3.1 Prinzipien

Jenseits der Details der konkreten Implementierung haben sich im Laufe der nun vierjährigen Projektarbeit einige grundlegenden Prinzipien herauskristallisiert, die als Leitlinien bei der Entwicklung der Datenbank Mehrsprachigkeit dienen

und von denen wir glauben, dass sie auch in anderen Zusammenhängen, in denen es um die computergestützte Verarbeitung heterogener linguistischer Daten geht, von Nutzen sein mögen. Diese Prinzipien sind im Folgenden zusammengefasst.

3.1.1 Drei-Ebenen-Architektur

Die Datenbank Mehrsprachigkeit geht von einer Drei-Ebenen-Architektur der computergestützten Datenverarbeitung aus:



Diese beinhaltet zunächst eine *Trennung von Form und Inhalt* von Dokumenten, die im Rahmen texttechnologischer Verfahren mittlerweile als selbstverständlich gilt, innerhalb der methodologischen Grundlagen linguistischer Transkription aber bislang noch weitestgehend unbeachtet geblieben ist. Die Form von Transkriptionen betrifft ihre graphische Darstellung, z.B. für eine gedruckte Ausgabe auf Papier oder für die Anzeige in einem Transkriptions- oder Auswertungswerkzeug auf dem Computerbildschirm. Viele Gegensätze bestehender Transkriptionsverfahren, beispielsweise die jeweils favorisierte Notationsform (z.B. Partitur- vs. Zeilennotation) oder die Darstellung spezifischer Gesprächsphänomene (z.B. „Pausenzeichen“), sind allein der Formebene zuzurechnen. Der Inhalt von Transkriptionen besteht hingegen aus einer Menge von symbolisch beschriebenen Gesprächseinheiten sowie deren Beziehungen zueinander. Relevante Unterschiede zwischen Transkriptionssystemen auf der Inhaltsebene be-

treffen beispielsweise die Definition und Benennung von Gesprächseinheiten (z.B. Äußerungen vs. Phrasierungseinheiten) und den Detailliertheitsgrad der Markierung von zeitlichen Relationen (s.o.). Durch eine konsequente Trennung von Form und Inhalt kann bereits eine wesentlich erhöhte Flexibilität im Umgang mit Transkriptionsdaten erzielt werden, denn sie ermöglicht es, ein und dasselbe Dokument für unterschiedliche Zwecke auf unterschiedliche Weise zu visualisieren. Darüber hinaus bietet sie die Möglichkeit, rein formbasierte Differenzen zwischen verschiedenen Transkriptionssystemen durch eine einheitliche Repräsentation auf der Inhaltsebene aufzuheben.

Über die Trennung von Form („Anwendungsebene") und Inhalt („Datenebene") hinaus sieht die Drei-Ebenen-Architektur eine weitere *Unterscheidung zwischen logischer und physikalischer Datenstruktur* vor. Die logische Struktur beschreibt unabhängig von konkreten technologischen Umgebungen die grundlegenden Organisationsprinzipien für Transkriptionsdaten in Form eines Datenmodells. Beispielsweise wird im Annotationsgraphen-Formalismus (Bird/Liberman 2001) vorgeschlagen, Transkriptionsdaten auf der logischen Ebene als azyklische gerichtete Graphen zu beschreiben, während das NITE-Object-Model (Evert et al. 2003) von einem System überlappender Hierarchien als grundlegender struktureller Organisationsform ausgeht. Auf der physikalischen Ebene hingegen wird festgelegt, wie diese abstrakten Datenstrukturen als computerlesbare Dateien zu kodieren sind. Wie Bird/Liberman (2001) feststellen, ist nur durch eine Trennung von logischer und physikalischer Datenebene sicherzustellen, dass Transkriptionsdaten über spezifische technologische Umgebungen hinaus langfristig nutzbar und austauschbar bleiben.[1]

[1] Gegenwärtig überdeckt der flächendeckende Einsatz von XML diesen Umstand häufig. Da XML oft nicht nur als Standard für die physikalische Repräsentation strukturierter Daten angesehen wird, sondern eng mit einem zugehörigen logischen (OHCO-)Datenmodell assoziiert ist, vernachlässigen einige aktuelle Ansätze diese essentielle Unterscheidung. Ge-

3.1.2 Aspekte der „Standardisierung“

Im Zusammenhang mit der computergestützten Verarbeitung heterogener Datenbestände wird oft deren „Standardisierung“ als grundlegendes Desiderat genannt. Die obigen Ausführungen zu konzeptuell vs. technologisch bedingter Heterogenität und zur Drei-Ebenen-Architektur der Datenverarbeitung können helfen, verschiedene Aspekte dieses Begriffs zu differenzieren:

Für diejenigen Unterschiede zwischen Daten, die sich aus konzeptuellen Überlegungen motivieren, verbietet sich eine Standardisierung. Weil unterschiedliche Forschungsziele und theoretische Hintergründe teilweise unterschiedliche Datenformen zwingend erfordern, kann das Ziel einer projektübergreifend einsetzbaren Lösung nicht sein, eine vollständig vereinheitlichte Form für Transkriptionsdaten vorzuschlagen. Vielmehr muss sich eine solche Lösung darauf beschränken, auf der Basis struktureller Gemeinsamkeiten verschiedener Systeme ein abstraktes „Framework“ zu erarbeiten, das möglichst wenige ontologische Festlegungen[2] trifft und für verschiedene theoretische Herangehensweisen parametrisierbar ist.

Während auf der logischen Ebene der Datenverarbeitung eine Standardisierung also auf ein solch abstraktes Framework begrenzt bleiben muss, bieten sich auf der physikalische Ebene weit reichende Möglichkeiten für die Nutzung von Standards: viele praktische Probleme in der Arbeit mit heterogenen Datenbeständen ergeben sich weniger aus deren prinzipieller konzeptueller Inkompatibilität, sondern vielmehr aus der Tatsache, dass sie in proprietären (binären oder textbasierten) und damit schwer zu verarbeitenden Formaten vorliegen. Der

---

rade für Transkriptionsdaten, deren grundlegende strukturelle Merkmale (insb. parallele Strukturen) sich nicht vollständig in das „Standard-XML-Datenmodell“ einordnen, ist es m.E. jedoch wichtig anzuerkennen, dass mit der Wahl von XML als Speicherformat noch keine Entscheidung über eine logische Datenstruktur getroffen ist.

[2] Auch Bird/Liberman (2001) bezeichnen ihren Annotationsgraphen-Ansatz als „ontologically parsimonious“.

einheitliche Einsatz von XML und Unicode kann daher auf dieser Ebene bereits zu einer entscheidenden Erleichterung der Verarbeitung beitragen.[3]

Auf der Anwendungsebene hingegen scheint eine Standardisierung weder wünschenswert noch notwendig. Eine Vielfalt von Darstellungsmöglichkeiten für Transkriptionsdaten kommt der Forschungspraxis ebenso entgegen wie die Möglichkeit, ein und dasselbe Datum mit unterschiedlichen Software-Werkzeugen bearbeiten zu können, und die Trennung von Form und Inhalt von Dokumenten stellt sicher, dass diese Vielfalt auf der Anwendungsebene keine Inkompatibilitäten auf der Datenebene nach sich ziehen muss.

3.1.3 Berücksichtigung bewährter Arbeitsweisen

Die Konstruktion der Datenbank Mehrsprachigkeit findet in einem Umfeld statt, in dem sich bereits viele verschiedene Ansätze für die computergestützte Verarbeitung von Transkriptionen gesprochener Sprache – teilweise über viele Jahre hinweg – etabliert haben. Zwar verspricht das Projektziel einen offensichtlichen qualitativen Sprung gegenüber all diesen Ansätzen; dennoch ist die Akzeptanz der entwickelten Lösungen in hohem Maße davon abhängig, dass bewährte Arbeitsweisen nicht übergangslos über Bord geworfen werden.

So müssen bereits beim Entwurf von Datenmodellen und -formaten Zugeständnisse an Unzulänglichkeiten älterer Datenbestände gemacht werden, denn deren Überführbarkeit stellt eine unabdingbare Voraussetzung für das Erreichen der Projektziele dar. Die Konvertierung von „Legacy Data" ist daher alles andere als eine triviale, rein technologische Aufgabe – sie führt notgedrungen zu ei-

[3] Tatsächlich ist nach unserer Erfahrung der Schritt von einem beliebigen textbasierten oder binären Format zu einem XML-basierten Format in der Regel um ein Vielfaches aufwändiger als eine Überführung eines XML-Datums in ein anderes XML-Datum. Da XML sich flächendeckend durchzusetzen scheint, steht zu hoffen, dass viele der momentan akuten Probleme im Umgang mit digitalen Sprachressourcen in Zukunft obsolet sein werden.

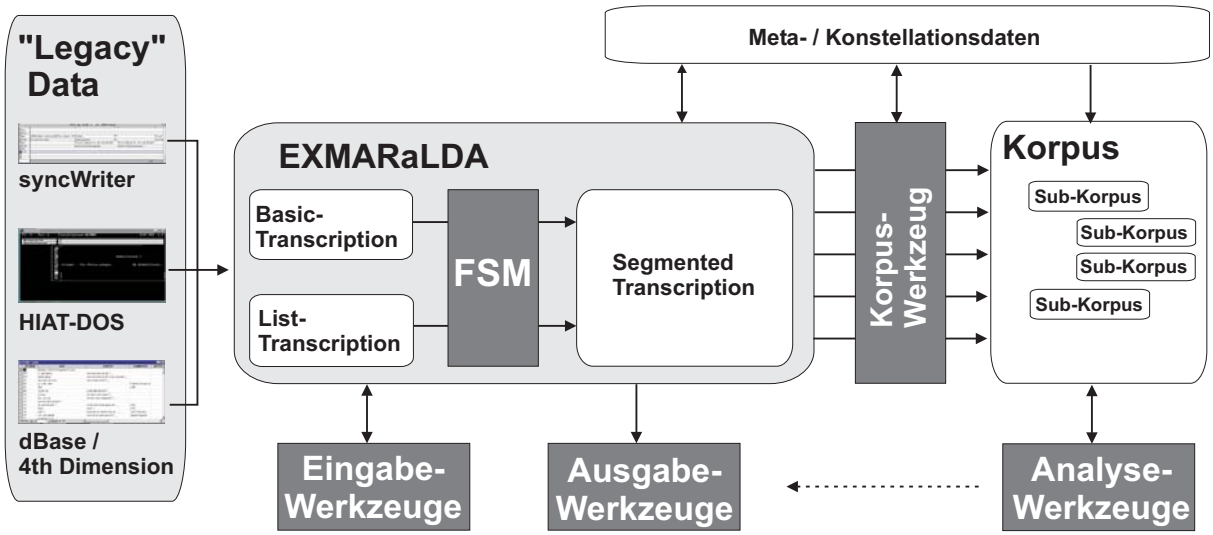
ner Reihe von Kompromissen und bestimmt so in entscheidender Weise die Architektur des zu entwickelnden Systems mit.

Unter den Aspekt der Beibehaltung bewährter Arbeitsweisen fällt auch die Berücksichtigung solcher Analyseschritte, die nur am Rande einer *computergestützten* Korpusarbeit zuzurechnen sind: so beinhalten diskursanalytische Verfahren als einen wichtigen methodischen Schritt eine intensive qualitative Analyse eines *gedruckten* Transkripts. Die üblicherweise bildschirmzentrierten Verfahren computergestützter Korpusarbeit müssen daher um die (insbesondere im Falle der Partiturnotation technisch anspruchsvolle) Möglichkeit der Ausgabe gedruckter Visualisierungen ergänzt werden.Schließlich hat die Orientierung an bewährten Arbeitsweisen auch dazu geführt, dass für die Datenbank Mehrsprachigkeit generell ein „Bottom-Up"-Konzept verfolgt wird, das eine verteilte Erstellung von einzelnen Transkriptionen und Korpora einer „zentralistischen" Datenverwaltung vorzieht. Die in vielen vergleichbaren Projekten angestrebte „Top-Down"-Lösung, nach der die Zusammenführung verschiedener Datenbestände in einer gemeinsamen Oberfläche ein übergeordnetes Ziel darstellt, hat sich für die Forschungspraxis am SFB 538 als nicht praktikabel erwiesen. Ausschlaggebend dafür waren einerseits Vorbehalte der einzelnen Forscher, die teilweise ihre Kontrolle über Persönlichkeits- und Urheberrechte bzgl. der Daten gefährdet sahen. Andererseits ließ auch die Prämisse, entwickelte Werkzeuge möglichst einfach Personen außerhalb des SFB zur Forschung und Lehre zur Verfügung stellen zu können, eine Lösung sinnvoll erscheinen, in der einzelne Bestandteile des Systems möglichst unabhängig von einer übergeordneten Architektur nutzbar sind.[4]

[4] Natürlich schließt eine solche „Bottom-Up"-Lösung nicht aus, dass dezentral erstellte Daten in übergeordneten Strukturen zusammengefasst und zugänglich gemacht werden. Sie beschränkt sich aber darauf, die *Voraussetzungen* für einen solchen Schritt zu schaffen und

3.2 Systemarchitektur

Folgende Abbildung illustriert die Systemarchitektur der Datenbank Mehrsprachigkeit:



Auf eine detaillierte Darstellung der Einzelkomponenten soll an dieser Stelle verzichtet werden. Statt dessen mögen die folgenden Ausführungen deutlich machen, wie diese Architektur mit den im vorigen Abschnitt angeführten Prinzipien zusammenhängt:

Die zentrale Komponente der Systemarchitektur ist das EXMARaLDA-Datenmodell. Dieses stellt eine eingeschränkte und spezifizierte Version des Bird/Libermannschen Annotationsgraphen-Datenmodells dar, geht also davon aus, dass sich Transkriptionsdaten angemessen als azyklische gerichtete Graphen auffassen lassen, deren Knoten den gemeinsamen Zeitbezug aller transkribierten Einheiten repräsentieren und deren Kanten die nicht-zeitlichen Informationen tragen. Diesem zeitbasierten Datenmodell auf der logischen Ebene entsprechen auf der physikalischen Ebene drei aufeinander aufbauende XML-

belässt die Entscheidung und Kontrolle über die konkrete Form einer übergeordneten Datenverwaltung bei den Einzelprojekten.

Formate: Die volle strukturelle Komplexität des Datenmodells kann in einer *Segmented-Transcription* repräsentiert werden. *Basic-Transcription* und *List-Transcription* bilden jeweils strukturell vereinfachte Untermengen einer solchen *Segmented-Transcription*. Die primären Eingabeinstrumente (insb. der Partitur-Editor) können der Effizienz halber zunächst auf diesen vereinfachten Untermengen operieren und nach Abschluss des Eingabeprozesses die Daten automatisch in das mächtigere *Segmented-Transcription*-Format überführen.[5] Die Überführung vorhandener Datenbestände besteht zunächst in der Abbildung von deren Strukturen auf das zeitbasierte Datenmodell und dann in der konkreten Konvertierung der jeweiligen Dateien in die entsprechenden XML-Formate.

Datenmodell und -formate sind gemäß den obigen Überlegungen prinzipiell unabhängig von Präsentationsformaten und Bearbeitungssoftware. Geeignete Ein- und Ausgabewerkzeuge werden teilweise im Projekt selbst entwickelt (EXMARaLDA Partitur-Editor), die Systemarchitektur sieht aber ausdrücklich vor, dass auch andernorts entwickelte Software, die auf ähnlichen Datenmodellen operiert (Praat, TASX-Annotator und ELAN), für die Erstellung von EXMARaLDA-Daten verwendet werden kann.

Zusammen mit solchen Ein- und Ausgabewerkzeugen bildet EXMARaLDA bereits ein selbstständig, d.h. ohne weitere übergeordnete Komponenten, nutzbares System und wird als solches auch vielfach in Forschung und Lehre eingesetzt. Um dem Ziel gerecht zu werden, eine gemeinsame Plattform für die verschiedenen SFB-Projekte zu bilden, kann es jedoch durch weitere Komponenten ergänzt werden: Ein Korpus-Werkzeug (CoMa, EXMARaLDA Corpus-Manager) erlaubt die Bündelung mehrerer EXMARaLDA-Transkriptionen zu

[5] Der in der Praxis hierfür benutzte Mechanismus ist eine Finite State Machine (FSM), die sich die Regelmäßigkeiten der verwendeten Transkriptionssysteme zunutze macht, um in den transkribierten Symbolketten implizite Markierungen in explizite Strukturrepräsentationen umzuwandeln.

einem Korpus, das wiederum in Form einer XML-Datei physikalisch repräsentiert wird. Ein in der Entwicklung befindliches Analysewerkzeug (SQUIRREL, Search and Query Instrument for EXMARaLDA) operiert dann auf Untermengen solcher Korpora, die anhand einer Suche auf Meta- und Konstellationsdaten (z.B. Sprechereigenschaften, verwendete Sprachen, Diskurstyp) ausgewählt werden.

4 Praktische Erfahrungen

Obwohl ein erheblicher Teil der sprachwissenschaftlichen Methoden sich heutzutage auf eine computergestützte Verarbeitung von Sprachkorpora stützt, und obwohl innerhalb der vergangenen fünfzehn Jahre eine Vielzahl entsprechender Werkzeuge und Datenformate entstanden ist, bleibt die Entwicklung von computergestützten Systemen für die linguistische Forschung ein Thema, das bislang kaum Eingang in die wissenschaftliche Literatur gefunden hat. Bei Beginn des Projekts „Datenbank Mehrsprachigkeit" bestanden demzufolge lediglich vage Vorstellungen über den zeitlichen und personellen Aufwand und die potentiellen inhaltlichen und technologischen Schwierigkeiten, die ein solches Vorhaben mit sich bringt. Im Laufe der nunmehr vierjährigen Projektarbeit haben sich diese Vorstellungen – in einem teilweise mühsamen Lernprozess – konkretisiert, und das Folgende ist der Versuch, einige der wesentlichen diesbezüglichen Erfahrungen festzuhalten.

4.1 Entwicklungsarbeit

4.1.1 Technologien

Hinsichtlich der zu verwendenden Technologien wurden bereits zu Projektbeginn drei verbindliche Entscheidungen getroffen:

Grundlage der physikalischen Datenrepräsentation sollten XML und Unicode sein, weil beide als offene Standards und aufgrund ihrer sich anbahnenden

Akzeptanz in der gesamten Internet-Welt einen Ausweg aus dem Problem der mangelnden Archivierbarkeit von Transkriptionsdaten versprachen. Dieses Versprechen ist eingehalten worden. XML- und Unicode-Technologie wird mittlerweile zuverlässig von einer Vielzahl von Werkzeugen und Programmierbibliotheken unterstützt, und der weitaus größte Teil vergleichbarer Projekte weltweit sieht ebenfalls XML- und Unicode-basierte Lösungen für die physikalische Repräsentation von digitalen Sprachdaten vor. Gegenüber der Ausgangssituation, in der die Vielfalt an proprietären (und teilweise kaum dokumentierten) Formaten und Kodierungen einfachste Verarbeitungsschritte oft unmöglich machte, stellt dies einen kaum zu überschätzenden Fortschritt dar. Weitere wesentliche Verbesserungen wären aus unserer Sicht vor allem dann zu erwarten, wenn auf XML aufbauende Technologien (insbesondere XSLT, XSL:FO, SMIL) einen der XML-Kerntechnologie vergleichbaren Grad der Unterstützung und Zuverlässigkeit erreichen würden.

Als Grundlage für die Implementierung der Software wurde JAVA ausgewählt. Damit verband sich vor allem die Erwartung, mit vertretbarem Aufwand Werkzeuge entwickeln zu können, die plattformübergreifend – insbesondere in der Windows- *und* der Macintosh-Welt – einsetzbar sind. Grundsätzlich ist auch diese Erwartung erfüllt worden – alle EXMARaLDA-Werkzeuge sind auf verschiedenen Betriebssystemen lauffähig –, allerdings hat sich die JAVA-Philosophie des „Write once, run anywhere“ stellenweise nicht bewahrheitet. Dies liegt einerseits darin begründet, dass die Macintosh-Implementierung der Java-Maschine bis heute unter gelegentlicher Instabilität und mangelnder Dokumentation leidet, was entsprechende fortwährende betriebssystemspezifische Anpassungen des Codes notwendig macht. Andererseits scheint besonders die im Zusammenhang mit der Transkription gesprochener Sprache wichtige Arbeit mit digitalisierten Medien-Signalen (Audio und Video) ein Teilbereich zu sein, der von „hardware-fernen“ Technologien wie JAVA prinzipiell nicht optimal

unterstützt werden kann.[6] Auf lange Sicht wünschenswert wären in dieser Hinsicht plattformspezifische Lösungen mit entsprechenden Interfaces zu JAVA.

4.1.2 Entwicklungsphasen / Zeitlicher Aufwand

Eine weitestgehend unbekannte Größe zu Projektbeginn war der für die einzelnen Phasen der Entwicklungsarbeit zu veranschlagende zeitliche Aufwand. Unterscheidet man nach der gängigen Praxis des Software-Engineering in etwa die folgenden Phasen der Entwicklungsarbeit – Planung/Analyse/Entwurf, Implementierung, Test, Dokumentation, Wartung, Benutzersupport –, so hat die Projektarbeit deutlich gezeigt, dass der zeitliche Aufwand für die „eigentliche" Programmierung (d.h. Implementierung und Wartung) der Software von den übrigen Größen um ein Vielfaches übertroffen wird.

Zu Projektbeginn gestaltete sich zunächst die Definition eines Anforderungsprofils für die zu entwickelnden Systemkomponenten sehr aufwändig. Die Erwartungen der potentiellen Benutzer beschränkten sich zunächst auf sehr allgemeine Anforderungen (wie „Benutzerfreundlichkeit der Software", „Hilfe bei quantitativen Analysen") und konnten auch in mehreren „Brainstorming"-Treffen nicht hinreichend spezifiziert werden. Es wurde daher zunächst auf der Basis einer vorläufigen Liste von wünschenswerten Merkmalen ein Prototyp eines Transkriptionseditors implementiert und interessierten Personen zur Verfügung gestellt. Dabei (und auch im folgenden Projektverlauf) zeigte sich, dass das Testen und kritische Begutachten von Beta-Software eine Tätigkeit ist, die einer Aufmerksamkeit und Sorgfalt bedarf, für die in der alltäglichen Forschungspraxis kaum Raum zu sein scheint – es erwies sich als sehr schwierig und zeitaufwändig, Forscher zur Auseinandersetzung mit einer Software zu be-

[6] Technologien wie das „Java Media Framework" und „Java Sound API" bieten zwar eine durchaus brauchbare Unterstützung für grundlegende Funktionen in dieser Hinsicht. Sie arbeiten nach unserem Eindruck jedoch merklich weniger präzise und zuverlässig als Lösungen, die in Sprachen wie C++ o.ä. implementiert wurden.

wegen, die aufgrund ihres frühen Entwicklungsstadiums keinen unmittelbaren praktischen Nutzen für die aktuell anstehende Forschungsarbeit zu versprechen vermochte. In diesem Sinne hat sich die ursprüngliche Erwartung, dass der allseits geäußerte dringende Bedarf an einer zeitgemäßen Transkriptions-Software von sich aus zu einer Vielzahl von Testern führen würde, als trügerisch erwiesen. Entscheidende Abhilfe schuf hier erst die Einstellung von Hilfskräften, die explizit mit dem Abfassen von Test-Berichten beauftragt wurden.

Nach diesen anfänglichen Schwierigkeiten hat sich inzwischen ein zirka drei- bis viermonatiger Zyklus etabliert, in dem neue Software-Versionen über die Projekt-Website veröffentlicht, anschließend Rückmeldungen über Bugs und Verbesserungsvorschläge gesammelt und diese in die Software eingearbeitet werden. Zu beobachten ist dabei, dass grundlegende Funktionserweiterungen wesentlich langsamer wahrgenommen werden als Umgestaltungen in bereits vorhandenen Komponenten. Die Zahl der „Power-User“, also von Personen, die neue Programmfunktionen in ihrem vollem Umfang frühzeitig und intensiv nutzen, ist vergleichsweise gering; der Großteil der Benutzer zeigt sich eher an der Optimierung von Vorhandenem interessiert. Der wichtigste zeitliche Faktor bei der Weiterentwicklung der Software ist aber mittlerweile weniger die Definition und Implementierung der Änderungen und Erweiterungen selbst, sondern deren Dokumentation in Form von Benutzerhandbüchern und Beispielen. Ähnliches gilt für die individuelle Beratung von Nutzern (vornehmlich über E-Mail), die einerseits zwar häufig nur die in der schriftlichen Dokumentation enthaltene Information dupliziert, anderseits aber auch entscheidend dazu beigetragen hat, dass mittlerweile eine wesentlich konkretere Vorstellung über den tatsächlichen und potentiellen Nutzerkreis der Software besteht.

4.2 Benutzer

Obwohl das Kernziel des Projektes in der Entwicklung einer Lösung für die Projektarbeit *am SFB 538* besteht, hat die EXMARaLDA-Software (insb. der Partitur-Editor) inzwischen eine recht weite Verbreitung über den SFB hinaus gefunden. Da bis vor kurzem der Download eine schriftliche Anmeldung voraussetzte (und über die Erfahrungen aus dem individuellen Benutzersupport, s.o), besteht zumindest eine ungefähre Vorstellung darüber, wie sich der derzeitige EXMARaLDA-Benutzerkreis zusammensetzt: Nach einer vorsichtigen Schätzung wurden seit Dezember 2001 (Version 1.0. des Editors) ca. 800 Benutzerkennungen angefordert. Weit über die Hälfte davon stammten von Studierenden, die EXMARaLDA im Rahmen einer sprachwissenschaftlichen Lehrveranstaltung nutzten, dies zum allergrößten Teil an deutschen Universitäten, teilweise aber auch im Ausland, vor allem in der Schweiz und den USA. Ebenfalls in der Lehre kommt EXMARaLDA bei der Lehramtsausbildung für Mathematiker und in der Kommunikationsforschung zum Einsatz. Projekte, die EXMARaLDA in der Forschung einsetzen, verfolgen zum überwiegenden Teil gesprächsanalytische Fragestellungen, weitere Anwendungsfelder finden sich in der Spracherwerbsforschung und in handlungsanalytisch orientierten erziehungswissenschaftlichen Projekten.

Wie bereits erwähnt, hat sich die aus dieser weiten Verbreitung resultierende hohe Zahl von kritischen Rückmeldungen bereits positiv auf die Entwicklungsarbeit ausgewirkt. Darüber hinaus erlaubt sie erste Mutmaßungen darüber, wie die Entwicklung computergestützter Systeme in der derzeitigen Forschungslandschaft aufgenommen wird. Zwei Aspekte scheinen mir hierbei besonders

wichtig, und das Folgende ist ein Versuch, diese in der gebotenen Kürze (und Vorsicht[7]) zu formulieren:

4.2.1 Die Rolle des Computers in der linguistischen Methode

Man kann den Einsatz des Computers für sprachwissenschaftliche Untersuchungen unter zwei Gesichtspunkten betrachten: zum einen kann der Computer als ein technisches Instrument gesehen werden, das vornehmlich dazu dient, gewisse Arbeitsschritte, die prinzipiell auch ohne seine Hilfe durchführbar wären, zu vereinfachen.[8] Zum anderen können computergestützte Verfahren aber auch als eine grundsätzliche Erweiterung des wissenschaftlichen Methodenrepertoires aufgefasst werden, etwa indem der Rechner als ein Instrument zum Anfertigen und Manipulieren wissenschaftlicher Modelle gesprochener Sprache betrachtet wird.[9] Nach unserer Erfahrung ist im Bereich der Gesprächs- und Spracherwerbsforschung, in denen EXMARaLDA vornehmlich zum Einsatz kommt, die erste Sicht die eindeutig vorherrschende. Der Nutzen neuer Lösungen wird weniger danach beurteilt, welche neuen Möglichkeiten sie bieten, sondern eher danach, wie sie bestehende Methoden zu unterstützen vermögen. Konkret äußert sich dies in der bereits angesprochenen Zurückhaltung der Nutzer beim Erpro-

---

[7] Die hierbei unvermeidlichen und eigentlich unzulässigen Verallgemeinerungen bitte ich nachzusehen. Dass sich diese subjektiven Eindrücke kaum „objektiv“ durch Verweise auf eine öffentliche wissenschaftliche Diskussion belegen lassen, ist Teil des Problems, das hier thematisiert werden soll.

[8] Beim Transkribieren betreffen solche Vereinfachungen z.B. die (auf dem Papier aufwändigeren) iterativen Korrekturschritte, das (auf dem Papier teure und u.U. nicht verlustfreie) Vervielfältigen und Verteilen von Transkripten, das (bei analogen Geräten oft umständliche und verschleißbehaftete) Abspielen der zu transkribierenden Aufnahme oder das (ohne Computerunterstützung mühselige und u.U. unzuverlässige) Suchen nach sprachlichen Phänomenen in größeren Korpora.

[9] Orlandi (2002) bringt diese unterschiedlichen Auffassungen wie folgt zum Ausdruck: “[Some] colleagues refer to the computer as ‘just a tool’ or ‘simply a bunch of techniques’, as if ways of knowing did not have much to do with what is known. Because the computer is a meta-instrument – a means of constructing virtual instruments or models of knowing – we need to understand the effects of modelling on the work we do as humanists.” Vgl. dazu auch Schmidt (i.V.)

ben solcher Funktionalitäten, die nicht bereits aus vorhandenen Systemen bekannt sind, aber auch darin, dass – bis auf wenige Ausnahmen – die Rolle computergestützter Verfahren in den methodologischen Grundlagen der genannten Gebiete bislang weitestgehend unreflektiert bleibt.[10]

4.2.2 Kooperative Korpusarbeit

Ein leitender Gedanke bei der Konstruktion der Datenbank Mehrsprachigkeit ist die Überwindung von Hindernissen, die einem projektübergreifenden Zugriff auf Korpora gesprochener Sprache derzeit im Wege stehen. Dies erscheint einerseits aus rein ökonomischen Gesichtspunkten wünschenswert, denn die Erstellung von Aufnahmen und Transkriptionen ist bekanntermaßen mit hohem finanziellem und personellem Aufwand verbunden, der sich umso eher rechtfertigen lässt, je mehr Forschende auf die solchermaßen entstandenen Ressourcen zugreifen können. Andererseits sprechen auch gegenstandstheoretische Gründe dafür, Korpusarbeit als eine kooperative Aufgabe wahrzunehmen, denn oft kann nur durch die Zusammenlegung verschiedener Korpora die „kritische Masse" erzielt werden, die für eine aussagekräftige (u.U. statistisch untermauerte) quantitative Analyse sprachlicher Phänomene notwendig ist.

Im Bereich der Sprachtechnologie hat diese Erkenntnis bereits zu einer ganzen Reihe von Initiativen und Institutionen geführt, die sich der Bereitstellung einer organisatorischen und technischen Infrastruktur für den Austausch von digitalen Sprachressourcen widmen.[11] Innerhalb der nicht technologisch ausgerichteten Linguistik wird die Zweckmäßigkeit solcher Bemühungen zwar nicht grundsätzlich in Frage gestellt; es hat bis heute aber weder eine nennens-

---

[10] Hingegen mangelt es nicht an Reflexionen über die Methode der Transkription als solcher. Gerade auf diesem Gebiet versprechen moderne texttechnologische Methoden – z.B. die o.g. Trennung von Inhalt und Form von Transkriptionen – aber eine grundlegende Erweiterung der Möglichkeiten etablierter Verfahren.

[11] Z.B. Organisationen wie ELDA oder LDC und Projekte wie EAGLES/ISLE oder ATLAS.

werte Anbindung an solche Initiativen stattgefunden, noch existieren eigenständige Konzepte, um die mancherorts entwickelten Einzellösungen (zu denen z.B. die CHILDES-Datenbank zählt) in praktikabler Weise miteinander zu verbinden. Projekte wie das hier vorgestellte (und weitere der auf dem Workshop präsentierten Arbeiten) können zwar gewisse Voraussetzungen für eine solche Infrastruktur schaffen, indem sie zumindest innerhalb der Institutionen, an denen sie angesiedelt sind, einen gemeinsamen Überbau für die Korpusarbeit entwerfen. Mindestens ebenso wichtig wäre jedoch eine dezidierte Bereitschaft der beteiligen Forscher, Fragen der Austauschbarkeit und Archivierbarkeit von Sprachdaten von vorneherein in die Korpuserstellung einzubeziehen, und die Entwicklung von Infrastrukturen, innerhalb derer Korpora anderen Interessierten zur Verfügung gestellt werden können, aktiv zu unterstützen. Die in Bird/Simons (2002) ausgeführte Beobachtung, dass diese Bereitschaft nicht uneingeschränkt gegeben ist, weil viele Forscher die Nachteile kooperativer Korpusarbeit höher bewerten als die sich aus ihr ergebenden Vorteile[12], können wir bestätigen.

5 Zusammenfassung und Ausblick

Wie andere vergleichbare Arbeiten zeigt auch das Beispiel der Datenbank Mehrsprachigkeit und EXMARaLDA, dass die Anwendung texttechnologischer Methoden und Konzepte und der Einsatz standardisierter und plattformübergreifend nutzbarer Technologien einen wesentlichen Fortschritt für die Arbeit mit heterogenen linguistischen Daten mit sich bringen kann. Das vornehmliche Ziel des vorliegenden Aufsatzes ist jedoch darauf hinzuweisen, dass dadurch nach unse-

[12] Dies betrifft ganz besonders die Frage der Zitierfähigkeit wissenschaftlicher Primärdaten. Bird/Simons (2002) sagen dazu: „Commonly, a researcher wants to derive recognition for the labor that went into creating primary language documentation, but does not want to make the materials available to others until deriving maximum personal benefit."

rer Erfahrung mindestens ebenso viele neue Fragen aufgeworfen wie alte beantwortet werden. Die nunmehr vierjährige Projektarbeit hat nämlich immer deutlicher werden lassen, dass technologische Innovation (sprich: Softwarewerkzeuge, Datenmodelle und -formate) nur eines von drei Standbeinen ist, auf die sich eine solcher Fortschritt stützt. Weitere entscheidende Verbesserungen sind zu erwarten, wenn die Möglichkeiten, die durch technische Weiterentwicklungen eröffnet werden, von einer entsprechenden methodologischen Reflexion begleitet und durch den Aufbau einer geeigneten Infrastruktur innerhalb der betroffenen Forschergemeinden unterstützt werden. Idealerweise würde dazu die Arbeitsteilung zwischen Texttechnologen und Informatikern, die Software und Datenformate entwickeln, und Sprachwissenschaftlern, die diese anwenden, teilweise aufgehoben oder zumindest stärker als bisher durch einen interdisziplinären Dialog ergänzt.

6 Literatur

Bird, Steven / Liberman, Mark (2001): *A formal framework for linguistic annotation.* In: Speech Communication 33(1,2), 23-60.

Bird, Steven / Simons, Gary (2002): *Seven Dimensions of Portability for Language Documentation and Description.* In: Language 79, 557-582.

Evert, Stefan / Carletta, Jean / O'Donnell, Timothy J. / Kilgour, Jonathan / Vögele, Andreas / Voormann, Holger (2003): The NITE Object Model. Version 2.1. (24 March 2003). NITE Internal document. http://www.ltg.ed.ac.uk/NITE/documents.html

Orlandi, Tito (2002): Is humanities computing a discipline? In: Braungart, Georg / Eibl, Karl / Jannidis, Fotis (Hrsg.) (2002): Jahrbuch für Computerphilologie 4. Paderborn: Mentis, 51-58.

Rehbein, Jochen / Schmidt, Thomas / Meyer, Bernd / Watzke, Franziska / Herkenrath, Annette (2004): Handbuch für das computergestützte Transkribieren nach HIAT. Arbeiten zur Mehrsprachigkeit, Serie B (56). Hamburg.

Schmidt, Thomas (2004): Transcribing and annotating spoken language with EXMARaLDA. In: Witt, Andreas / Heid, Ulrich / Carletta, Jean / Thompson, Henry S. / Wittenburg, Peter (Hrsg.): XML-based richly annotated corpora. LREC 2004 Satellite Workshop. Paris: ELRA, 69-74.

Schmidt, Thomas (i.V.): Computergestützte Transkription als Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln. Dissertation, Universität Dortmund.

Thomas Schmidt
Universität Hamburg
SFB 538 „Mehrsprachigkeit“
Max Brauer-Allee 60
22765 Hamburg
Germany
thomas.schmidt@uni-hamburg.de
http://www.rrz.uni-hamburg.de/exmaralda

Heterogeneity and Standardization in Data, Use, and Annotation: a Diachronic Corpus of German*

Anke Lüdeling
Humboldt-Universität zu Berlin

This paper describes the standardization problems that come up in a diachronic corpus: it has to cope with differing standards with regard to diplomaticity, annotation, and header information. Such highly heterogeneous texts must be standardized to allow for comparative research without (too much) loss of information.

1 Introduction

Most of the corpora in linguistics are fairly large corpora of modern languages (or language stages) that are characterized by a high degree of standardization in three areas: (a) the input data, (b) the annotation, and (c) the intended use.

Input Data In modern languages/language stages orthography is standardized, and texts of the same text type adhere to certain conventions or rules, which makes these texts similar to each other. Most of the tools and mechanisms for collection, pre-processing and evaluation of corpora—symbolic or quantitative—exploit such regularities.

Annotation Most corpora are also standardized with respect to the annotation—be it header information, positional annotation, or structural annotation. Standardization efforts like TEI, OLAC, or IMDI[1] concentrate on these layers.

* I want to thank Stefanie Dipper, Lukas Faulstich, Michael Götze, Ulf Leser, Thorwald Poschenrieder, the DDD project members, and the audience of the Potsdam workshop for valuable comments.

Interdisciplinary Studies on Information Structure 02 (2005): 43-54
Dipper, S., M. Götze and M. Stede (eds.):
Heterogeneity in Focus: Creating and Using Linguistic Databases

Intended Use Corpora are usually collected with a given goal in mind and are tailored to fit that goal. This is true even of seemingly 'multi-purpose' corpora like the British National Corpus or the American National Corpus, which are collected specifically for synchronic linguistic research.

On the other hand, we have many less 'well-behaved' corpora: corpora of less studied languages with sometimes little orthographic standardization, or corpus collections that encompass different languages and annotation needs that do not easily conform to corpus-linguistic standards.

The goal of this paper is the description of the problems arising in standardizing the highly variable and heterogeneous data for a diachronic corpus of German. I point to solution strategies and approaches for the representation of the data.

In the following section, I describe the characteristics of our corpus DEUTSCHDIACHRONDIGITAL, a diachronic corpus of German. In Sec. 3, I argue that in addition to a maximally flexible corpus architecture and data model we need a 'corpus model' that ensures standardization and homogeneity wherever possible.

2 DeutschDiachronDigital

The project DEUTSCHDIACHRONDIGITAL[2] (henceforth DDD) aims at the collection, annotation, and presentation of a diachronic corpus of German, covering

[1] The URLs and references for all corpora, tools and other resources mentioned in this paper are given in Section 5.2.

[2] DeutschDiachronDigital is a Germany-wide interdisciplinary project with 12 partners (universities and research institutions). It is still in its beginning phase, with the final funding decision pending. The architecture was developed in a preparatory project funded by the Senatsverwaltung für Wissenschaft, Forschung und Kultur, Berlin. For more information refer to http://www.deutschdiachrondigital.de/.

the language stages Old High German (OHG), Old Saxon (Old Low German; OS), Middle High German (MHG), Early Modern German (EMG), Middle Low German (MLG), and Modern German (MG). DDD thus contains texts from about 800 to about 1900 AD, the focus being on the older texts. The corpus is designed in such a way that it can be used by scholars from (historical) linguistics, (historical) philology, lexicography, history etc. This means that it must be possible to annotate and search for very different kinds of information. Possible research questions include: (i) how did the meaning or form of a specific word change, (ii) how did a given syntactic construction change, (iii) what case do the arguments of a given verb feature, (iv) how did a given genre (let's say, the novel) evolve, (v) what did a given author say about some philosophical concept, (vi) how do letters written by women in the 17th century differ from letters written by men?

At the moment there are quite a number of digitized texts and corpora of older language stages of German, mostly collected in small individual projects with differing standards with regard to diplomaticity, annotation, and header information. Because of these differences, it is at present not possible to conduct qualitative or quantitative research across more than one language stage. A further obstacle is that coverage of the languages stages is very different (for MHG, for example, there are relatively large balanced corpora while for EMG there are almost no electronic resources available, see the survey of Kroyman et al. 2004 for details).

In the following section I describe different kinds of variation within the data and present some ideas of how DDD will cope with them.

3 Standardization Problems and Methods of Resolution

There are in principle two different strategies to deal with the diversity of the data:

(1) normalization or categorization (that is loss of information)

(2) the preservation of different readings[3] (either by explicit coding or by underspecification).

The DDD project uses both strategies, for different types of problems.

3.1 Corpus Architecture

The problem of combining different kinds of texts and annotations in one corpus or database has been tackled in a number of projects (see EXMARaLDA, TUSNELDA, ANNIS etc.; for an overview see Dipper et al. 2004b). Many of them achieve high flexibility by following stand-off models (first developed in multi-modal corpora, see for example Carletta et al. 2003), where the annotation of a text is independent of the text itself and therefore even conflicting hierarchies in different levels of annotation can be accommodated. In some of these tools, the individual texts in the corpus may have differing annotation levels that are in principle totally independent of each other. The flip side of this flexibility is often lack of standardization so that these corpus collections are simply that: collections of texts with no common properties.[4]

---

[3] I use the term 'readings' to refer to differences in form or meaning, not just to semantic differences.

[4] This is, of course, due to the situations in which these environments are developed—typically large research groups (Sonderforschungsbereiche) which work on one specific research question in many different languages and approaches; for them it is not a requirement that these resources be directly comparable.

Another problem in conjunction with systems like EXMARaLDA, TUSNELDA, or ANNIS is that they are solutions to very specific problems and cannot easily be transferred to other situations.

For the DDD corpus architecture we use a multi-layer stand-off corpus design with a diplomatic text version as the timeline. Thus, our architecture is as flexible as EXMARaLDA etc.: new texts and new annotation layers can be added at any time. The corpus is stored in a central relational database, import and export to different XML formats is provided via web-clients. The data model is based on an ODAG (ordered directed acyclic graphs) model (see Carletta et al. 2003). Details of the DDD corpus architecture are described in Dipper et al. (2004a) and Faulstich, Leser & Lüdeling (2005). In the remainder of this paper I want to focus on standardization.

3.2 Input Data: Non-Standardized & Multilingual Data

DDD is a historical and diachronic corpus. Historical corpora, even if they consist of texts of one period only, have to deal with non-standardized texts. Not only are there no or little orthographic conventions (depending on the age of the text), there are also many special characters, abbreviations, etc. that are particular to one text. For some historical periods of German (e.g. MHG) it has long been customary to normalize in editions and textbooks. Normalization has a number of advantages: it facilitates readability, and eases access for lexicographical purposes, etc. However, it 'throws away' information about spelling differences and paleographic specifics. Most existing historical corpora designed for linguistic purposes normalize to a certain extent (e.g. the HELSINKI CORPUS) or digitize from already normalized editions of the text instead of original manuscripts or early prints (Rissanen et al. 1993). An exception is the MENOTA PROJECT, which aims at high diplomaticity and is therefore well-suited for paleographical research (however, because corpus composition is not standardized it is less suited for linguistic or lexicographical questions).

Instead of opting for one or the other variant, DDD's multi-layer architecture refers to a highly diplomatic version of the text as a sort of time line and aligns a semi-diplomatic version.

Often there is more than one witness (manuscript, copy) for a given text. DDD digitizes from originals (manuscripts or early prints) rather than critical editions and the focus is on representativity. Hence, it is often necessary to pick one manuscript out of several candidates. In some special cases, however, we will digitize two or more witnesses of the same text (for example, two manuscripts or a manuscript and a critical edition). In these cases we treat each text as a separate text, which comes with its own annotation layers. The witnesses can then be aligned.

Besides being a historic corpus, DDD is a diachronic corpus. A diachronic corpus can be seen as a multilingual corpus (with some parallel portions due to texts that exist across different language stages, such as biblical texts). In addition to standardization on one level, a multilingual corpus has to deal with standardization across different levels, which causes standardization problems in particular at the level of annotation.

3.3 Annotation

Most texts will be annotated with basic structural information, part of speech, lemma, and inflectional morphology. The most relevant standardization problems arise from the fact that tags in any tag set will change their denotation over time. For example, the properties of what would be classified as an 'adverb' are not stable from OHG to MG. This will be dealt with in two ways: the tag sets will be built up hierarchically so that information can be left underspecified if necessary. It will also be possible to explicitly code alternatives.

Standardizing the annotation of lemmas causes particular problems. Ideally, lemmas should be standardized within each language level. Because of the orthographic variance (there are at least 17 spellings of the lemma *und* 'and' in MHG: *undi, unnti, vnnti, vnte*, ...), this is difficult. For some language stages (especially for MHG), there is a well-established normalizing tradition that can be adopted. For other stages, standards have to be developed; cf. the situation in OHG: it is customary to base the normalization on the lemmas as they occur in the largest available OHG text, 'Tatian'. However, this means that many lemmas (namely all those that do not occur in Tatian's text) have to be made up 'in the way Tatian would have written them'. Poschenrieder (2004) suggests a different way of normalizing by representing different but related sounds by abstract 'hyper letters'.

There are no conventions for lemma correspondence that hold across all language stages. Lexical change occurs on the semantic, morphological, and stratic level (or combinations thereof, see Gévaudan 2002 and Gévaudan & Wiebel 2004 for a discussion and a modelling proposal). Therefore, it is not a trivial task to decide which elements (i.e., normalized lemmas) correspond to each other across language stages.

3.4 Intended Use: Corpus Composition

As stated above, DDD will be used by scholars from different fields such as (historical) linguists, (historical) philologists, lexicographers, historians etc. as well as by interested laypeople. In this way it already differs from most available corpora. The existing historical and diachronic corpora are usually compiled either for linguistic purposes or for special philological, lexicographic, or historical purposes (for an overview over historical and diachronic corpora see Kroymann et al. 2004). This is reflected by the *corpus composition*: historical corpora for linguistics or lexicography, such as the HELSINKI CORPUS or the cor-

pora for the MITTELHOCHDEUTSCHES WÖRTERBUCH, are in some way 'representative', e.g., they cover language stages, authors, genres, etc. in given proportions (Klein 1991, Biber 1993). Corpora for philological purposes, on the other hand, are often much more specialized—they cover the work of one author only, or sometimes even only one work (the CANTERBURY TALES PROJECT), or one genre (LANCASTER NEWSBOOK CORPUS), for an overview see Burch et al. (2003).

To make diachronic research possible, a diachronic corpus must be 'representative' with respect to time, dialect, text type, etc. This is difficult to achieve in diachronic corpora because categories like 'text type' or 'dialect' are not stable across time. Some genres or text types[5] only develop during the sampling time (like the category 'novel') and others appear and then disappear again (like minne songs). Even if the categories were stable and one could draw a matrix of different parameters, it would not be possible to fill all the cells in such a matrix because there is simply not enough material (this is, of course, especially true for the early stages).

The DDD project deals with these problems in two ways: a very detailed common set of parameters (time, genre, dialect, information about the author, register, purpose of the text etc.) is chosen, which is represented in hierarchies so that it is possible to always use the most specific category. The categories form a matrix, and corpus selection aims at filling as many of the matrix cells as possible (many will remain empty). If there are several texts that fit a cell, the most well-known is chosen.

For the early language stages (OHG, OS), all available texts are included in the corpus. From the time of MHG/MLG on, there are too many texts avail-

[5] The problem of defining a 'genre' or 'text type' is ignored here. I assume that there is some kind of available definition.

able so that there has to be a selection. For even later language stages (older Modern German), the matrix would become too complex since many more genres develop. Therefore DDD concentrates on three genres (letters, newspaper text, novels) and leaves other cells empty (they can be filled later).

The categories and their values are represented in a TEI-conform header so that it is possible to construct sub-corpora for different purposes.

4 Conclusion

In order to make a corpus out of very different texts from different language stages, a maximally flexible corpus architecture is necessary as well as standardization in many ways. In a diachronic corpus, it must be possible to make use of as much information as possible from every text (because there are so few texts to begin with).

DDD deals with this situation by using a multi-layer architecture where text versions of different degrees of diplomaticity are aligned (see Lüdeling, Poschenrieder & Faulstich 2005). The tag sets are hierarchically ordered so that information can be left underspecified where necessary. In addition, different hypotheses can be coded explicitly. This architecture is able to cope with many of the challenges arising from the specific needs of diachronic data.

5 References

5.1 Bibliography

Biber, Douglas (1993) Representativeness in Corpus Design. In: *Literary and Linguistic Computing* 8, 243–257.

Burch, Thomas; Johannes Fournier; Kurt Gärtner & Andrea Rapp (eds.) (2003): *Standards und Methoden der Volltextdigitalisierung. Beiträge des Inter-*

nationalen Kolloquiums an der Universität Trier, 8./9. Oktober 2001. Akademie der Wissenschaften und der Literatur, Mainz.

Carletta, Jean; Jonathan Kilgour; Timothy O'Donnell; Stefan Evert & Holger Voormann (2003) The NITE object model library for handling structured linguistic annotation on multimodal data sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web*, Budapest.

Dipper, Stefanie; Lukas Faulstich; Ulf Leser & Anke Lüdeling (2004a) Challenges in Modelling a Richly Annotated Diachronic corpus of German. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, , pp. 21–29, Lisbon.

Dipper, Stefanie; Michael Götze & Manfred Stede (2004b) Simple Annotation Tools for Complex Annotation Tasks: an Evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, pp. 54–62, Lisbon.

Faulstich, Lukas; Leser, Ulf & Lüdeling, Anke (2005) *Storing and Querying Historical Texts in a Database*. Technical Report 176 of the Institut für Informatik. Humboldt-Universität zu Berlin. Available online at http://www.deutschdiachrondigital.de/publikationen/index.php

Gévaudan, Paul (2002) *Klassifikation des lexikalischen Wandels. Semantische, morphologische und stratische Filiation*, Dissertation, Universität Tübingen.

Gévaudan, Paul & Dirk Wiebel (2004) Dynamic lexicographic data modelling. A diachronic dictionary development report. In *Proceedings of the LREC Conference*, Lisbon.

Klein, Thomas (1991) Zur Frage der Korpusbildung und zur computergestützten grammatischen Auswertung mittelhochdeutscher Qellen. In: Wegera, Klaus-Peter (ed.) *Mittelhochdeutsche Grammatik als Aufgabe*. Zeitschrift für deutsche Philologie 110 (Sonderheft) 1991, 3–23.

Kroymann, Emil; Sebastian Thiebes; Anke Lüdeling & Ulf Leser (2004) *Eine vergleichende Analyse von historischen und diachronen digitalen Korpora*. To appear as Technical Report of the Institut für Informatik, Humboldt-Universität zu Berlin. Available online at http://www.deutschdiachrondigital.de/publikationen/index.php.

Lüdeling, Anke; Thorwald Poschenrieder & Lukas Faulstich (2005) DeutschDiachronDigital. Ein diachrones Korpus des Deutschen. In *Jahrbuch für Computerphilologie 2004*. Available online at http://www.deutschdiachrondigital.de/publikationen/index.php

Poschenrieder, Thorwald (2004) *Digitalisierung altgermanischer Texte*. Paper given at the Conference of the Society for Indo-European Studies, Krakau, http://www.deutschdiachrondigital.de/publikationen/index.php

Rissanen, Matti; Merja Kytö & Minna Pallander (1993) *Early English in the Computer Age: Explorations through the Helsinki Corpus*, Mouton de Gruyter, Berlin.

5.2 Corpora and Tools[6]

- ANNIS (a Linguistic Database for Exploring Information Structure): http://www.sfb632.uni-potsdam.de/annis/
- The CANTERBURY TALES PROJECT: http://www.cta.dmu.ac.uk/projects/ctp/
- EXMARaLDA (Extensible Markup Language for Discourse Annotation): http://www.rrz.uni-hamburg.de/exmaralda/
- The Diachronic Part of the HELSINKI CORPUS, delivered by ICAME, user manual: http://khnt.hit.uib.no/icame/manuals/HC/INDEX.HTM
- IMDI (Isle Metadata Initiative): http://www.mpi.nl/IMDI/
- The LANCASTER NEWSBOOK CORPUS:

[6] The URLs given below were valid URLs on March 11, 2005.

http://bowland-files.lancs.ac.uk/newsbooks/project.htm

- MITTELHOCHDEUTSCHES WÖRTERBUCH: http://gaer27.uni-trier.de/MWV-online/MWV-online.html
- The MENOTA PROJECT: http://gandalf.aksis.uib.no/menota/
- NITE http://nite.nis.sdu.dk/aboutNite/
- OLAC (Open Language Archives Community): http://www.language-archives.org/
- TEI (Text Encoding Initiative): http://www.tei-c.org/
- TUSNELDA (Tübinger Sammlung nutzbarer empirischer, linguistischer Datenstrukturen): http://www.sfb441.uni-tuebingen.de/tusnelda.html

Anke Lüdeling
Korpuslinguistik
Institut für deutsche Sprache und Linguistik
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin
Germany
anke.luedeling@rz.hu-berlin.de
http://www.linguistik.hu-berlin.de/korpuslinguistik/

Multiple Hierarchies: New Aspects of an Old Solution[1*]

Andreas Witt
Universität Bielefeld

In this paper, we present the Multiple Annotation approach, which solves two problems: the problem of annotating overlapping structures, and the problem that occurs when documents should be annotated according to different, possibly heterogeneous tag sets. This approach has many advantages: it is based on XML, the modeling of alternative annotations is possible, each level can be viewed separately, and new levels can be added at any time. The files can be regarded as an interrelated unit, with the text serving as the implicit link. Two representations of the information contained in the multiple files (one in Prolog and one in XML) are described. These representations serve as a base for several applications.

1 Introduction

Markup expresses characteristics or interpretation of text. It is obvious that there is, at least potentially, more than one view for a given text. Often it is necessary to express these different or alternative views of text explicitly, i.e. by markup. At the moment, it seems to be a tendency to annotate more and more information. This development definitely takes place in the field of linguistics, where language data is associated with information from several linguistic levels of description, e.g. semantics, syntax, morphology, phonology – levels which

---

[1] This paper is a slightly modified reprint. (Originally published in the Online-Proceedings of the Extreme Markup Languages 2004, see http://www.extrememarkup.com).

[*] The different aspects of this approach are used within several projects of 'Research Group: Text-technological Modeling of Information' which is funded by the German Research Foundation (DFG). I would like to thank Harald Lüngen and Neill Kipp for their help and all the reviewers of this paper for their helpful comments.

Interdisciplinary Studies on Information Structure 02 (2005): 55–85
Dipper, S., M. Götze and M. Stede (eds.):
Heterogeneity in Focus: Creating and Using Linguistic Databases

are (relatively) independent of each other. But also text simply published on the web is combined with more and more meta-information. Since markup expresses meta-information about text, the amount of markup will increase, especially if the semantic web will emerge. And, of course, more markup implies that it becomes more likely to encounter multiple hierarchies.

This paper deals with two different problems:

1. the problem of annotating overlapping structures, and
2. the problem that occurs when documents should be annotated according to different, possibly heterogeneous tag sets.

As a solution of both problems the technique of annotating documents in multiple forms is proposed and described in detail. The paper also discusses the disadvantages of the approach, disadvantages that are definitely the reason why a lot of projects reject this solution: “An obvious and also simple solution would be to make a separate file for each transcription. However, this makes comparison between levels unnecessarily cumbersome, and it is notoriously difficult to keep track of revisions in parallel files.” (Haugen, 2004)

This paper shows how it is possible and what is needed to overcome these problems.

2 Multi-hierarchically Structured Text

Publishing, especially print publishing, was the driving force behind the development of markup languages. Text was viewed as an *ordered hierarchy of content objects* (OHCO). Consequently most markup languages are based on the OHCO assumption. The term and the acronym were introduced by DeRose et al. (1990) and were further discussed by Renear et al. (1996).

2.1 Problems of OHCO-based Markup-Languages and Possible Solutions

From a formal point of view, SGML-based markup systems allow for the representation of exactly one hierarchy. Hence, in principle, only one structure can be represented in one document. In practice, this restriction often does not receive special attention as different structures often can be expressed within one hierarchy. Thus, e.g., the logical structure of a text, i.e. the division into captions, lists, sections etc., differs completely from the syntactic structure such as the division of the text into sentences and phrases. Especially, none of the elements belonging to the different tag sets overlap. Hence, it is possible to project both structures into one hierarchy without problems. The disadvantage is, however, that this necessarily results in a mixture of these structures, in the annotated text as well as in the corresponding document grammar.

The problem of multiple hierarchies is often discussed. The main reason for this might be the view of document engineers, who are faced with the fact that ranges of text marked up by SGML or XML elements must not overlap. Single-hierarchically structured text is a consequence of this restriction. If overlapping does not occur, the problem of combining heterogeneous tag sets is often ignored. Hence, a mixture of structures can be found quite often in text represented in one syntactic hierarchy. One example was already given, another example is HTML. Even in its ‘strict’ version, different structures can be mixed, at least through the often promoted use of the elements `span` and `div` combined with an assignment of a `class` information.

To avoid confusion when talking about multiply structured text and text ideally organized by multiple hierarchies, the terms ‘level’ or ‘level of description’ are used when referring to a logical unit, e.g. visual document structure or logical text structure. When referring to a structure organizing the text technically in a hierarchically ordered way, the terms ‘layer’ or ‘tier’ are

used. A level can be expressed by means of one or more layers and a layer may/can include markup information on one or more levels (cf. Bayerl et al., 1999).

2.1.1 SGML/XML Approaches

The problem of representing multiple hierarchies has often been addressed and several solutions have been proposed, especially in the field of humanities computing, which is by nature concerned with text and its interpretation or its description. Consequently, the best collection of techniques is presented by the *Text Encoding Initiative* (TEI, see ACH/ACL/ALLC (1994) and Barnard et al. (1995)). The TEI describes the techniques for using SGML for annotating multiple hierarchies. (1) `CONCUR`: an optional feature of SGML (not available in XML) which allows multiple hierarchies to be marked up concurrently in the same document, (2) milestone elements: empty elements which mark the boundaries between elements, in a non-nesting structure, (3) fragmentation of an item: the division of what logically is a single element into two or more parts, each of which nests properly within its context, (4) virtual joins: the recreation of a virtual element from fragments of text, (5) redundant encoding of information in multiple forms.

With the exception of the extremely rarely implemented option `CONCUR`, in effect, all of these techniques are workarounds:

- Milestones do not allow for making use of a key concept of XML, namely elements containing a range of text. This leads to several consequences:
    - No content model restriction can be stated by a document grammar for the range of text between the milestones marking the begin and the end of the region. This results in not being able to use an XML editor for annotating these regions.

    - Standard SGML parsers cannot check whether milestone elements marking the begin and the end of a region match.
    - It is more difficult or impossible to process these regions by means of a style sheet, e.g. by XSLT or, respectively, by CSS.
- The technique of fragmentation results in ‘containers’ containing only a part of the text. So for instance, an element `sentence` or `para` that is fragmented simply does not contain a sentence or a paragraph.
- The technique of virtual joins requires a separate interpretation of the SGML document.
- Redundant encoding in multiple forms results in multiple files which are not integrated in a larger unit containing all the information of the different layers.

Another technique not mentioned directly by the TEI guidelines is stand-off annotation, i.e. (new) layers of annotation are added by building a new tree whose nodes are SGML elements which do not contain textual content (`#PCDATA` in terms of the DTD syntax), but links to another layer.

In some respects stand-off annotation is a generalization of virtual joins, because not only contents of elements are joined, but also ranges between points within the document. Sometimes these ranges make use of markup already contained in a layer, sometimes special pointers are used to refer to the specific text elements which are the object of the annotation (Pianta and Bentivogli, 2004). With the first introduction of this concept (Thompson and McKelvie, 1997) this second approach was described.

In practice, however, most often an already-annotated layer is taken as the primary annotation tier, to which the stand-off annotation is linked. In the case of linguistic annotation often the annotation level ‘word’ is used as the primary annotation layer. In most of its applications, stand-off annotation makes use of

one layer as the link target of the new tier, but it is also possible to link to several already existing layers (see Carletta et al., 2003).

In any case, stand-off annotation results in new hierarchies established by new annotation layers that are linked to already existing annotations. Sometimes the new layer is included in the same document, sometimes the layers are separated.

This approach has the advantage that it is based on SGML/XML and that different levels of description are kept separate. However, this approach has some drawbacks too:

- The new layers require a separate interpretation.
- The layers, although separate, depend on each other. They can only be interpreted by reference to the layer(s) they point to.
- Although all information is included, the information is difficult to access using generic methods. As a consequence, standard parsing or editing software cannot be employed.
- Standard document grammars (e.g. the TEI Relax NG scheme, the XHTML-DTD, or the W3C Schema for DocBook) can only be used for levels containing both markup and textual data.
- Linking to a sub-element range, or to textual data not annotated at all is difficult. The pointing mechanism defined by the TEI or by XPointer can be used, but requires another special software solution.
- The primary layer should be a (primary) level. The choice of such a primary level is not an easy task. Often its declaration is arbitrary and artificial.

Despite these disadvantages the technique of stand-off annotation is used in a lot of projects faced with the problem of multiple hierarchies, especially in the area of annotating linguistic data.

2.1.2 Namespaces

The Namespace standard provides a mechanism to specify where a specific element has been defined (Bray et al. 1999). Connecting elements with their defining document grammars is done by adding a prefix to the element or the attribute names. The prefix points, at least conceptually, to a document grammar, in which the element or the attribute is defined. Thus the logical structure of a text can be marked up with e.g. XHTML elements for captions, sections, lists etc. and its syntactic structure can be marked up by using an adequate module of the DTD of the TEI. If a corresponding namespace has been defined, a caption belonging to the logical structure of the text can be referenced by `html:h2` instead of only `h2`, whereas a word or a morph can be marked up by `tei:w` or `tei:m` instead of `w` or `m`. This enrichment of the annotation facilitates the recognition of the relation between the annotation and a specific level (here text structure and morphology).

Unfortunately, some problems remain. Sometimes a document grammar defines several different structures, possibly in a modular way. The document grammars defined by the TEI-DTD are a good example of this. As an ad-hoc solution, one could try to define different namespaces for the same document grammar. A first prefix `teins1` and a second prefix `teins2` could be defined. Because the prefixes have only the function of a place holder for the expanded name spaces, it is necessary to declare several different ‘real’ namespaces for one DTD. But this would definitely be against the intention of the standard.

Nonetheless namespaces are an important help when using markup that belongs to different levels of description, since it provides a means to refer to an element not only by its name or its generic identifier but additionally by its defining document grammar.

A minor problem of namespaces might occur when using schema languages which allow for context-sensitive definitions of content models. With this technique it is possible to define a different content model for regions marked up with elements with the same element name. For example, Relax NG and XML Schema allow for such definitions. The (slightly) different definitions of an element `para` in sections and `para` in the context footnote, where (embedded) footnotes should be prohibited, is an often used example of the use of this option. But since the namespace points to the document grammar and not to the element definition, context-sensitively defined elements cannot be distinguished.

One problem has not been addressed by the namespace recommendation at all: the problem of overlapping hierarchies.

2.1.3 Non SGML-based Markup languages

Some non-SGML-based markup languages have been proposed in the last few years. An example of such a markup language is the Multi-Element Code System (MECS, Sperberg-McQueen and Huitfeld 1999) or TexMECS (Huitfeldt and Sperberg-McQueen, 2001). Its major extension with respect to SGML and XML is that overlapping ranges are admitted within documents.

In 2002 another markup language was proposed, called *Layered Markup and Annotation Language* (LMNL, Tennison and Piez (2002)). LMNL is a markup language which not only allows for annotating overlapping elements but also for connecting the element names to corresponding annotation levels. All structures modeled by XML can also be modeled by LMNL.

2.1.4 Discussion

The problem of annotating multiple hierarchies can be divided into two different and relatively independent problems: (1) SGML-based markup systems cannot handle ‘overlapping hierarchies’ and (2) the tag sets used or needed for a certain

annotation task are sometimes quite heterogeneous. The first problem is addressed by the solutions proposed in the TEI guidelines, by stand-off annotation, and by the TexMECS markup language, which does not conform to SGML. The second problem is addressed by the namespace recommendation.

LMNL provides a solution for both problems: regions marked up by different elements may overlap and its layered annotation approach is specially designed for this task. But, since LMNL does not conform to SGML, not to mention XML, it has not been applied up to now.[2]

Another possibility mentioned above is redundant encoding in multiple forms. This approach is rarely used by the markup community. The reasons for this seem to be clear: First, most people try to avoid redundancy. Second, and more important, multiple encodings in different forms are independent of each other, but people who deal with annotated text are only interested in an integrated format.

On the other hand, it is also an advantage if one annotated document is not related to another document, because then the document is an independent unit of information. This leads to several more advantages.

- If a document is used for separate annotation levels, this results in each level being able to be viewed separately and new levels to be added at any time, without reference to and dependence on existing files.
- Standardized document grammars can be used for some annotation levels and specialized document grammars can be defined in an intuitive way, i.e. declaring that an element can contain text and not only attributes whose values point to some other element in some other annotation layer.

[2] One exception is described by Alexander Czmiel (2004). He implemented a subset of LMNL in an XML-conformant way. Of course, some of the advantages of LMNL cannot be achieved by such an XML-based representation.

Moreover, the approach (as well as stand-off) has additional advantages over the milestones and the fragmentation approach:

- The modeling of alternative annotations based on different theoretical assumptions is possible (see Sasaki et al. (2003) for the usefulness of this point in the field of linguistics).
- Each document instance uses its own DTD (or Schema), i.e. document grammars are not mixed up.

We therefore conclude that this approach has a lot of advantages with respect to the aspects of editing, maintenance, interchange, and reusability of XML-annotated data. What remains to be solved is the main drawback of independent annotations: How is it possible to connect these layers?

We also conclude that a special representation model for these data is needed, because of the redundancy in the data. This representation format is desired for storing and processing this information. From a theoretical point of view, LMNL would be an ideal format. From a practical viewpoint a stand-off annotation approach is most suited for these tasks and, in fact, is used most frequently.

2.2 Multiple Annotations and their Representation

Beside the advantages of the annotation in multiple form, the main problem of this approach has been addressed: the independence of the tiers. But interrelations of annotation layers are of interest for many persons concerned with structuring and modeling of information. In this section a method is presented which complements the advantages of *redundant encoding of information in multiple forms* with possibilities to link these multiple forms and represent them uniformly. Furthermore, conversion tools for the annotation format and possible representation formats are described.

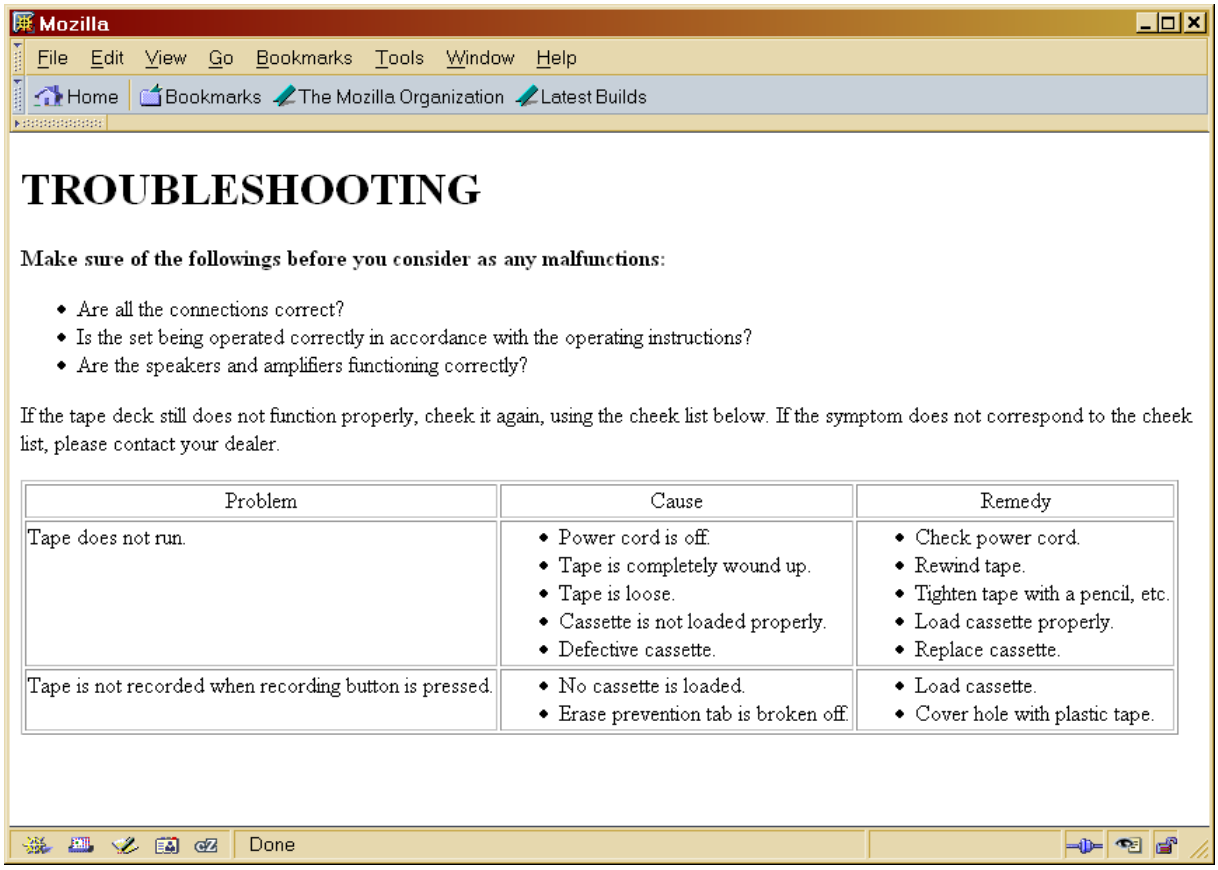


Fig. 1: Screenshot of the rendering of the HTML-version of the example-text

2.2.1 XML-based Multi-layer Annotation

One obvious way to interrelate different annotations of same textual data exists. The different annotations could be regarded as transformations of each other. Hence, the relations between the XML documents can be declared in an XSLT-program or an XSLT-stylesheet. This stylesheet can be viewed as a description of relations between two XML vocabularies. But for composing such a stylesheet it is necessary to have information on the relation of the elements defined in the different vocabularies. Moreover, this approach could only be successful, if the relations between the elements can be stated unambiguously.

Another way to link the *different forms* was proposed by Witt (2002). The central idea of this approach is that the annotated text itself serves as the link. This is achieved by annotating exactly the same text several times.

This approach is described by means of a simple example. Below the XHTML-source of a user's manual is given (see also Fig. 1)

```
<xhtml><h1>TROUBLESHOOTING</h1>
...
<table border="1">
 <tr>
  <td align="center">Problem</td>
  <td align="center">Cause</td>
  <td align="center">Remedy</td>
 </tr>
 <tr>
  <td valign="top">Tape does not run.</td>
  <td valign="top"><ul>
   <li>Power cord is off.</li>
   <li>Tape is completely wound up.</li>
   <li>Tape is loose.</li>
   <li>Cassette is not loaded properly.</li>
   <li>Defective cassette.</li>
 </ul></td>
 <td valign="top"><ul>
   <li>Check power cord.</li>
   <li>Rewind tape.</li>
   <li>Tighten tape with a pencil, etc.</li>
   <li>Load cassette properly.</li>
   <li>Replace cassette.</li></ul></td>
</tr>
<tr>
 <td valign="top">Tape is not recorded when recording button
is pressed.</td>
 <td valign="top"><ul>
   <li>No cassette is loaded.</li>
   <li>Erase prevention tab is broken off.</li>
 </ul></td>
 <td valign="top"><ul>
   <li>Load cassette.</li>
   <li>Cover hole with plastic tape.</li></ul>
 </td>
</tr>
</table></xhtml>
```

The same fragment of text can be annotated in a more content-oriented way or semantically:

```
<r><h1>TROUBLESHOOTING</h1>
...
<p-c-r>
 <description>
  <first>Problem</first>
  <second>Cause</second>
  <third>Remedy</third>
 </description>
 <case>
  <problem>Tape does not run.</problem>
  <potential_causes>
   <cause>Power cord is off.</cause>
   <cause>Tape is completely wound up.</cause>
   <cause>Tape is loose.</cause>
   <cause>Cassette is not loaded properly.</cause>
   <cause>Defective cassette.</cause>
 </potential_causes>
 <potential_remedies>
   <remedy>Check power cord.</remedy>
   <remedy>Rewind tape.</remedy>
   <remedy>Tighten tape with a pencil, etc.</remedy>
   <remedy>Load cassette properly.</remedy>
   <remedy>Replace cassette.</remedy></potential_remedies>
</case>
<case><problem>Tape is not recorded when recording button is
pressed.</problem>
 <potential_causes>
   <cause>No cassette is loaded.</cause>
   <cause>Erase prevention tab is broken off.</cause>
 </potential_causes>
 <potential_remedies>
   <remedy>Load cassette.</remedy>
   <remedy>Cover hole with plastic tape.</remedy>
 </potential_remedies>
</case>
</p-c-r></r>
```

As can be seen, the text content of both versions is identical, but the markup is different.

2.2.2 Representation

The multiply annotated XML documents are the basis of the representations. For further processing of the text it is necessary to represent them uniformly. Two alternative representations are described in the next subsections.

PROLOG

Sperberg-McQueen et al. (2001) discuss the *meaning and interpretation of markup*. For explaining their approach, annotated documents are represented in the programming language Prolog. In their representation, every element, attribute, and the content are saved as so-called Prolog facts. This approach has been extended, so that multiple annotations as described in the previous section can be represented. Through this all separate annotations can be associated in a data basis, which then can be used e.g. for automatic detection of relations between the annotation levels (see section 3.2).

In the simplest setting, for any element, attribute and text node of each annotation level a Prolog fact is built which contains the following information:

1. a cross reference to the annotation level;
2. the absolute start position of the text passage which is marked up;
3. the end position of that text passage;
4. the position of the unit in the tree representation of the annotation level;
5. the element name or — if necessary — the attribute name, respectively

Some Prolog facts containing information from the two levels of the above examples should serve as an illustration.

```
node('tape-xhtml.xml', 729, 786, [1,5,3,2], element('td')).
node('tape-xhtml.xml', 729, 786, [1,5,3,2,1], element('ul')).
node('tape-xhtml.xml', 729, 751, [1,5,3,2,…], element('li')).
node('tape-thema.xml', 729, 786, [1,5,3,2], element('pot…')).
node('tape-thema.xml', 729, 751, [1,5,3,…], element('cause')).
```

The first argument contains the name of a layer, i.e. `tape-xhtml.xml` and `tape-thema.xml`. The second element points to the beginning of a range annotated with the respective element (the fifth argument). In the example, all the ranges start at the same position. The end of each range is given as the third

argument. The position in the tree (argument four[3]) is given as a list, pointing to the nodes within the tree representation of the respective annotation layer.

Attributes are represented in a similar way, using the Prolog predicate `attr`:

```
attr('tape-xhtml.xml', 729, 786, [1, 5, 3, 2],
     'valign', 'top').
```

The textual content is given by the predicate `pcdata_node`:

```
pcdata_node(729, 730, 'N').
pcdata_node(730, 731, 'o').
pcdata_node(731, 732, ' ').
pcdata_node(732, 733, 'c').
pcdata_node(733, 734, 'a').
pcdata_node(734, 735, 's').
pcdata_node(735, 736, 's').
```

Such a collection of Prolog facts contains all the information of the different annotations and can serve as a data basis for further developments of Prolog programs.

XML-Based Representation

Multiply annotated XML files can also be represented in an XML-based format. Such a presentation could be achieved by transforming the Prolog facts into XML elements, e.g. the predicate `node` with its five arguments could be transformed to an empty XML element `node` with five attributes. However, such a *Prolog-in-XML* representation would not make much sense.

A representation using the technique of virtual joins, or stand-off annotation, is more interesting, because this technique is used to represent multiple hierarchies. Moreover, most of the above mentioned disadvantages of

[3] In first case this means: The element `td` is the second daughter of the third daughter of the fifth daughter of the root element.

this technique do not exist when this format is an add-on for the multiple annotation of XML layers.

The European language technology project NITE developed a format for representing heavily annotated data. This format is well suited for this task.

The NITE-format (Carletta et al., 2003) combines several files forming a corpus. These files are interrelated with each other. One way to represent the two annotation layers `tape-xhtml.xml` and `tape-thema.xml` is given in the next examples. The NITE-corpus consists of four separate files, in the examples these could be:

- `tape.corpus.xml` contains meta-information, e.g. names of the files of the corpus, names of the defined elements and attributes etc.
- `o1.stream.xml` contains the textual data supplemented with reference points for linking with the other layers
- `o1.tape-xhtml.xml` comprises the markup of `tape-xhtml..xml`
- `o1.tape-thema.xml` expresses the information provided by the markup of the file `tape-thema.xml`

One possible representation of the textual stream would supply any character with an ID:

```
<char nite:id="char_727">e</char>
<char nite:id="char_728">d</char>
<char nite:id="char_729">.</char>
<char nite:id="char_730">N</char>
<char nite:id="char_731">o</char>
<char nite:id="char_732"> </char>
<char nite:id="char_733">c</char>
<char nite:id="char_734">a</char>
<char nite:id="char_735">s</char>
<char nite:id="char_736">s</char>
```

Alternatively, in larger text single words could serve as the reference units.

The next example shows how the elements of the thematic annotation are linked to the text.

```
  <nite:child href="o1.stream.xml#id('char_727')" />
  <nite:child href="o1.stream.xml#id('char_728')" />
  <nite:child href="o1.stream.xml#id('char_729')" />
</problem>
<potential_causes nite:id="potential_causes_2" >
  <cause nite:id="cause_6" >
    <nite:child href="o1.stream.xml#id('char_730')" />
    <nite:child href="o1.stream.xml#id('char_731')" />
    <nite:child href="o1.stream.xml#id('char_732')" />
    <nite:child href="o1.stream.xml#id('char_733')" />
```

The elements `potential_causes` and `cause` begin at the character with the reference `char_730`, i.e. the first character of the string 'No cassette is loaded'. The string itself is given by references to the characters in the file `o1.stream.xml`.

2.2.3 Conversion

The conversion from XML to Prolog is implemented in Python. The program `xml2prolog.py` receives as an input one or more XML documents and outputs a collection of Prolog facts.[4]

The element `<Root>` is represented as the fact:

```
node(AnnotationLayer, 0, n, [1], element(Root)).
```

where `n` refers to the last character in the textual data. The XML attributes of the root element `att1` and `att2` and their values `val1` and `val2` are represented as two facts:

```
attr(AnnotationLayer, 0, n, [1], 'att1', 'val1').
attr(AnnotationLayer, 0, n, [1], 'att2', 'val2').
```

This representation contains some redundant information, because the pointers to the character (0 and n) could be inferred automatically by means of the

[4] This program is mainly written and maintained by Daniel Naber and Oliver Schonefeld. It is available via the project Web pages (http://www.text-technology.de; 'Projekt Sekimo').

information of the respective element, but the explicit indication of this information can speed up processing.

Some options for the transformation process are:

compare: the primary data, i.e. the `PCDATA` content of the elements of the XML files is compared; if the primary data is not identical, the first different character is shown;

pcdata/pcdatanodes: character data is included;

aggressive: whitespace is added or removed anywhere in the document if whitespace is the reason for differences of the primary data;

filter: some elements in some files should be filtered (including their textual content), e.g. `<script>` within HTML-documents.

That way it is possible to convert any number of identical but differently marked up texts into a collection of Prolog facts.

For the conversion of text which is annotated in multiple forms according to the NITE-format, another program has been developed.[5] This program is called `nexus.pl` and is implemented in the Perl programming language. The functionalities are similar to `xml2prolog.py`. The input is *n* annotations of the same text. The program outputs a NITE-corpus that consists of the *n+2* files described above.

2.2.4 Discussion

It has been shown that the technique of annotating the same text in multiple forms has many advantages and that its main drawback can be avoided. However, exactly the same data has to be annotated several times. With this prerequisite the multiply annotated files can be regarded as a unit which is heavily interrelated, because the text serves as the implicit link.

---

[5] This program has been developed by Jan Frederik Maas. Also this program is available via the project Web pages.

After that, two different formats have been described. One format is an interrelated Prolog representation of the information contained in the multiple files. The other format is based on XML and was developed for the processing and the exchange of linguistic corpora annotated on several levels of description.

Furthermore, programs for the automatic transformation of multiply annotated text to the integrated formats have been introduced.

3 Aspects of Processing Multiply Annotated Text

In this section, techniques and software implementations for editing, inferring and unifying separately annotated texts are presented. Moreover, a technique of unifying the multiple forms will be discussed.

3.1 Editing

The editing of copies of text, each annotated separately, definitely is not an easy task. One way to do this is annotating each file with the help of a standard XML editor. Since, at least in some scenarios, the text is given and need not be changed, this approach offers at least two advantages: standard XML-editing software is available and the automatic comparison of the textual content (e.g. by the option ‘compare’ of the transformation program `xml2prolog` described above) allows quality assurance, since it is highly unlikely that exactly the same modification of the textual data occurred twice (or even more times) in different files. Unfortunately, this has also several drawbacks. One of these is connected with the comparison of whitespace. Since sometimes whitespace matters, it makes no sense to collapse all whitespace. On the other hand, most often this difference should be ignored. Therefore a special whitespace normalization

program has been implemented.[6] But if textual data must be changed, textual content must be changed in different files. This task requires special editing software.

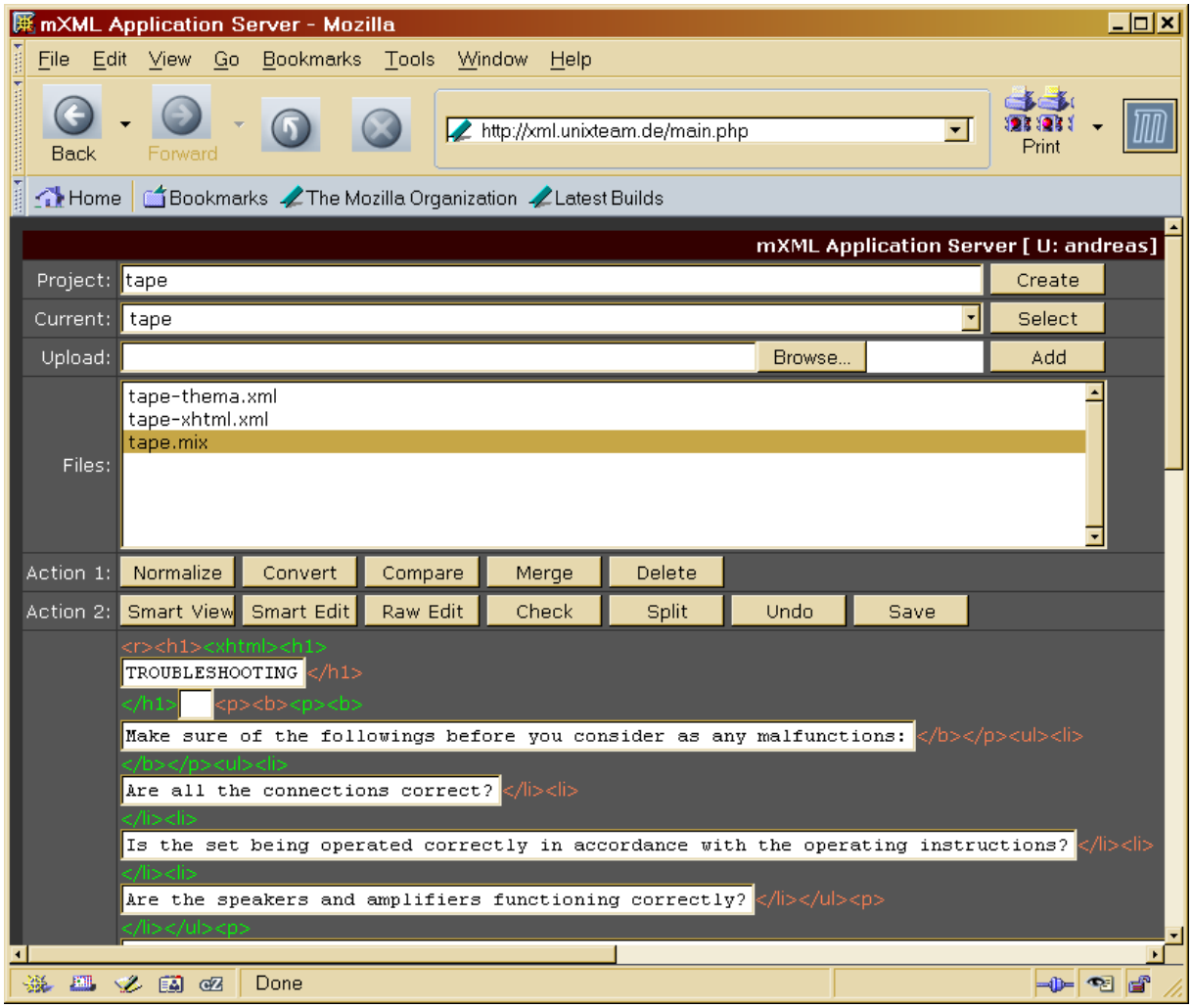


Fig. 2: Editor mode for changing textual content

At the time of writing this paper two master's thesis projects are concerned with implementing special editing software for this task.

One editor is web-based (implemented in PHP) and allows for typing and changing the textual content of multiply annotated files. The two screenshots

[6] This program is written and maintained by Oliver Schonefeld. It is available via the project Web pages (http://www.text-technology.de; 'Projekt Sekimo').

give an impression of this program. Fig. 2 shows how text can be modified. As can be seen, the markup cannot be changed in this mode.

Fig. 3 shows the non-XML-based markup employed internally by the editor. This format can be used by experts to modify not only the textual content but also the markup.

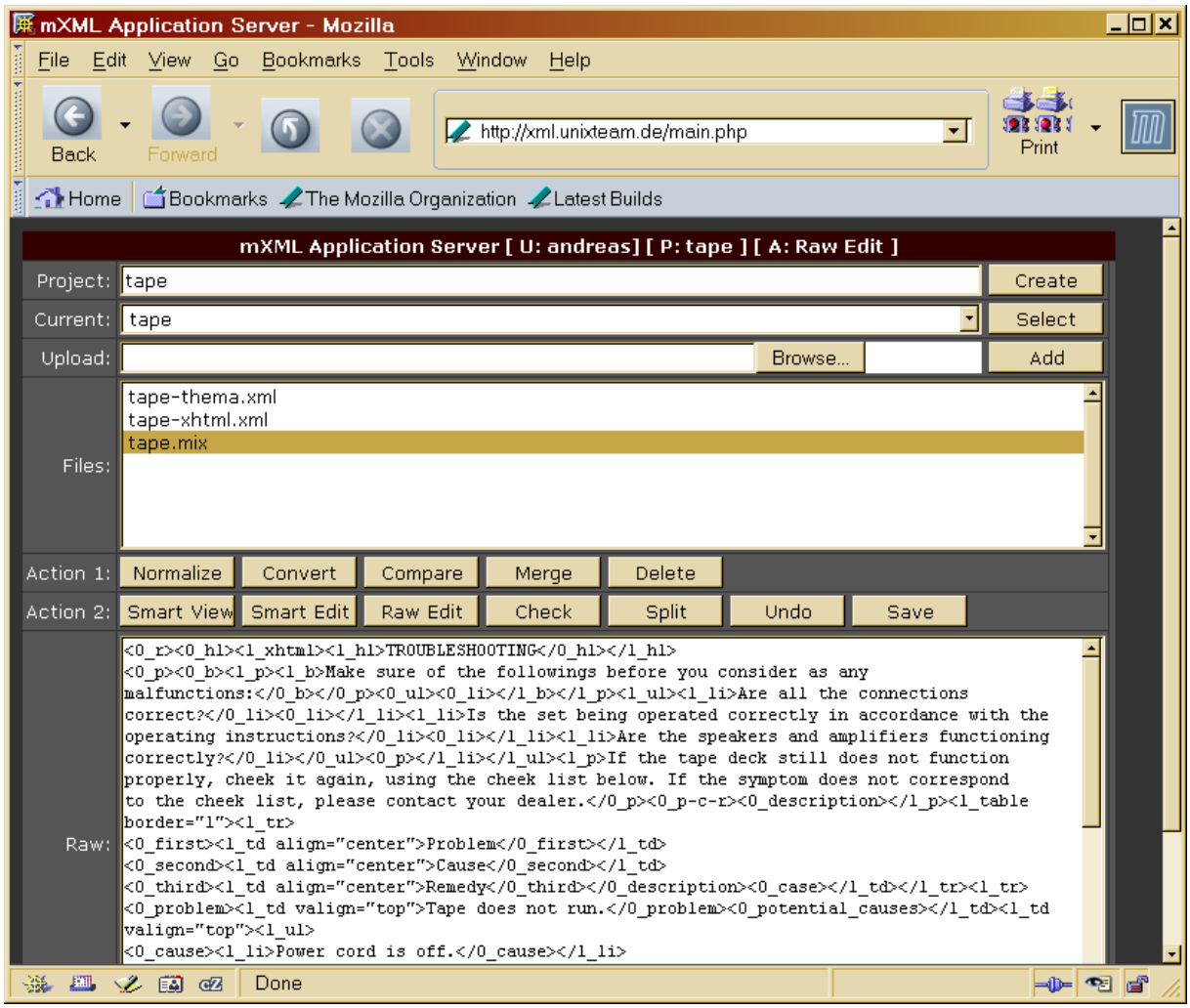


Fig. 3: Editor mode editing textual content and markup

The existence of such an editor is important for this approach. Otherwise, it is very difficult to change multiply annotated text, because each modification of the text must be done in each layer.

In a second master's thesis an editor will be implemented in the Java programming language, using the Eclipse platform. The aim of this master's

project is the implementation of an editor capable of associating several document grammars with one text. The insertion of elements is a two step process: first, the annotator refers to a document grammar the element belongs to and, second, (s)he can choose an element out of a list of elements that are allowed at this point according to the schema. When saving the document, for each associated schema one file will be saved. The validation will take place for each of these files.

3.1.1 Relations Between Annotations

The markup within a single document is hierarchically structured. The structure, leaving aside cross-references, can be represented as a tree. Between the nodes of the tree there exist certain relations, i.e. subordination, (direct) neighborhood, etc. These relations can be used for queries for structural characteristics in one layer. Such queries can be formulated in several ways, as e.g. with XSLT, in query languages like XQuery or (when using the appropriate library) in Prolog (see Sperberg-McQueen et al., 2002).

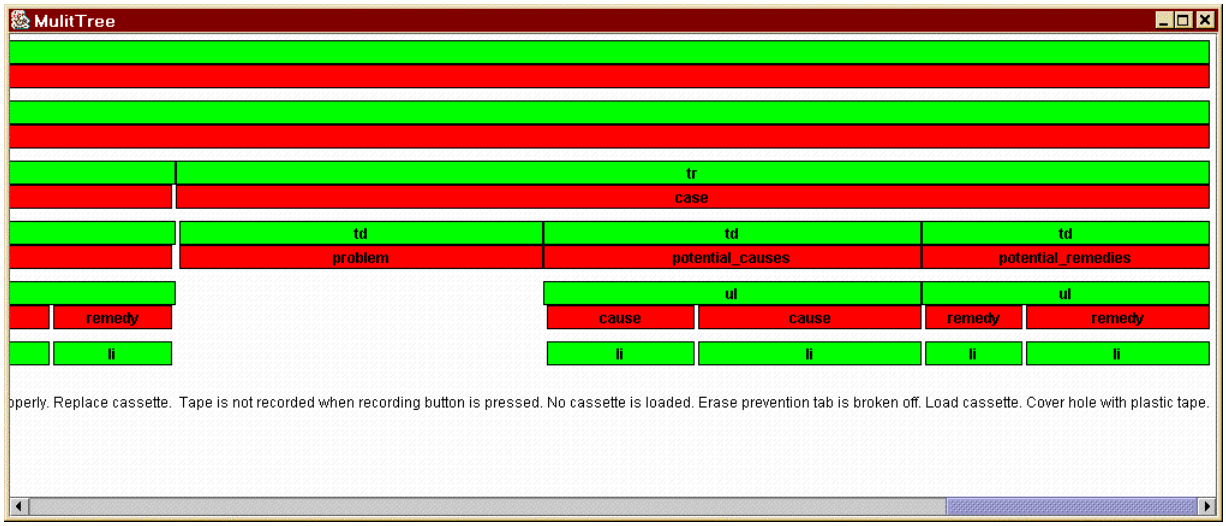


Fig. 4: Example annotation with two layers

When regarding more than one annotated layer more relations can be found. The figure above depicts the two layers of the example annotation, the XHTML

layer and the content-oriented annotation layer. This visualization shows some of these relations.

An aligned representation of both layers shows that an identical range in the primary data is marked up with different elements.

```
...<potential_causes><cause>No cassette is loaded.</cause>...
...<td valign="top"><ul><li>No cassette is loaded.</li>...
```

Durand (1999) and Durusau & O'Donnell (2002) assembled all the possible relations between elements of different layers. The visualization is based on the presentation of Durusau & O'Donnell (2002).

Start-tag identity

```
<a>.................................</a>
<b>............</b>
```

Full inclusion

```
<a>.................................</a>
           <b>.........</b>
```

Total identity

```
<a>.................................</a>
<b>.................................</b>
```

End-point identity

```
            <a>.....................</a>
<b>.................................</b>
```

Ranges annotated by different elements overlap

```
<a>.....................</a>
              <b>............................</b>
```

The end-position of one element is shared by the start-tag of another element

```
<a>.................</a>
                    <b>................</b>
```

etc.

Within our project, the Prolog fact base is used as a base for inferencing these relations. For this task, special Prolog predicates have been implemented.[7]

Alternatively, the NITE XML search tools[8] could be used for representations conforming to the NITE representation.

3.2 Relations Between Annotation Layers

More general information on the relations between element classes, i.e. the set of all instances of an element, is more interesting than a comparison of relations between single element instances. To do this, a set of meta relations have been defined. A meta relation holds under certain conditions.

The meta relation `identity` between the element classes `a` and `b` holds, if for every occurrence of an element instance `a` the same range of text is annotated by an element instance `b` and vice versa.

```
Meta-relation identity:
<a>.....................</a>
<b>.....................</b>
```

The meta relation `inclusion` between the element classes `a` and `b` holds, if for every occurrence of an element instance `a` the same range of text is annotated by an element instance `b`, and if the meta-relation `identity` does not hold, i.e. for all occurrences, one of the following configurations can be found:

```
<a>.................</a>
<b>..............................</b>

                          <a>.....................</a>
<b>.........................................</b>
```

[7] This program was mainly written by Daniela Goecke. It is available via the project Web pages.

[8] NXT Search is freely available (binaries, documentation, and source code) via http://www.ims.uni-stuttgart.de/projekte/nite/download.shtml.

```
        <a>.....................</a>
<b>.....................................</b>

<a>.....................</a>
<b>.....................</b>
```

The meta-relation `overlap` between the element classes `a` and `b` holds, if for every occurrence of an element instance `a` the range annotated by `a` overlaps with the range annotated by an element instance `b`. For all occurrences of `a`, the following configuration can be found:

```
<a>.....................</a>
           <b>.....................</b>
```

The inferred meta-relations indicate whether theoretical constructs modeled by (certain elements of) two document grammars are in some relation to each other. So it might be investigated whether certain constructs used by different linguistic theories (e.g. in traditional Japanese grammar and in ‘modern’ phrase structure grammars) are alphabetical variants of each other. Moreover, with these meta-relations, generalizations stated by researchers or inferred automatically on a small empirical basis can be falsified.

Unfortunately, however, the research conducted by the projects of the DFG research group mentioned above showed that these meta-relations do not hold very often. The reason for this lies in the way they are defined: a meta relation between two elements holds if certain conditions hold for *all* occurrences of these elements. It would be interesting to explore whether certain meta relations exist under certain conditions.

One possibility for a refinement of the meta relations is a description of specific contexts where these relations do hold. Context specifications allow for expressing such a condition.

A context specification could be expressed by a set of XPath expressions, but XPath seems to be a language that is too powerful for context specifications.

Therefore, an alternative format to express structural properties, called "Context Specification Document" (CSD), has been developed (Sasaki and Pönninghaus, 2003).

3.3 Unification of Annotation Layers

Of course, sometimes an integrated XML representation is necessary. Therefore a program for the unification of multiply annotated documents has been developed.[9] With this Prolog program two document layers can be merged. The architecture of this program is visualized in the next figure.

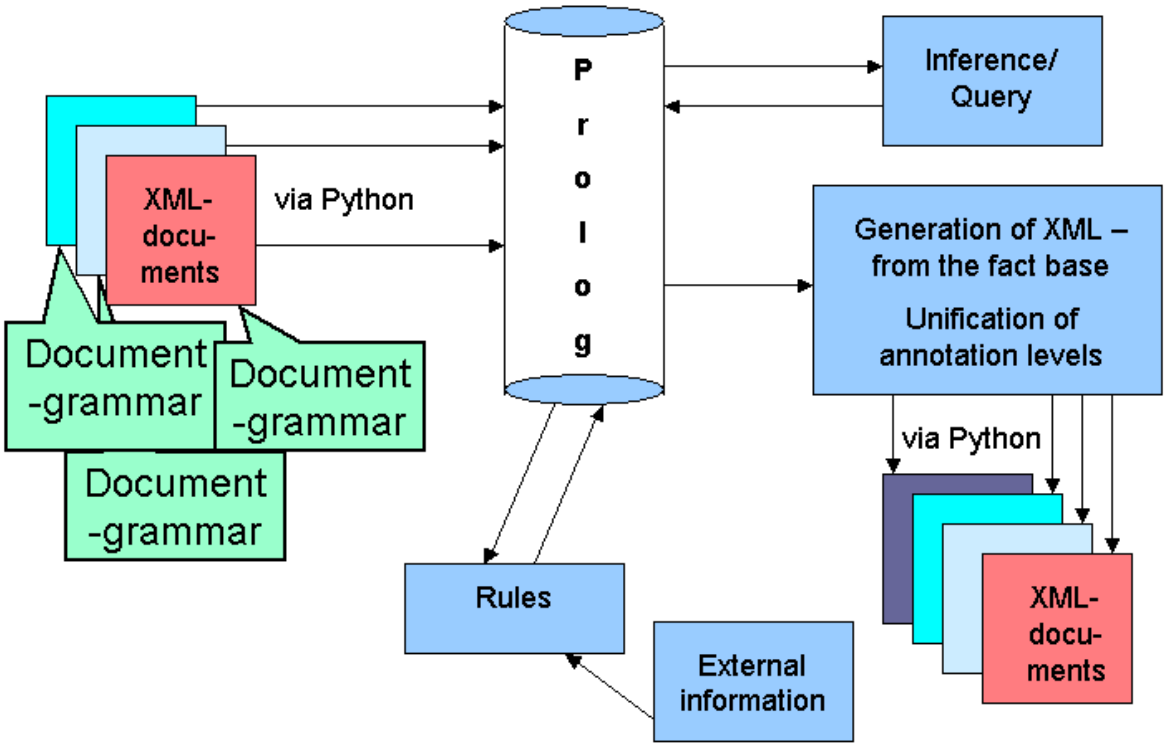


Fig. 5: Unification of annotation layers

[9] This program was mainly written by Daniela Goecke and is maintained by Harald Lüngen. It is called `semt.pl` and it is also available via the project web pages. It is also described by Witt et al. (2004).

The Prolog predicate (`semt`) receives four arguments:

- layer1 (to be unified)
- layer2 (to be unified)
- list of elements which should be deleted in the process of unification

The result of the merger (again a collection of Prolog facts) is written to a new file specified in the fourth argument. The new database contains a copy of all layers in the input database plus the result layer.

In case the unification results in a layer where the elements are not properly nested, a second result layer (a difference list) is created. The resulting database is re-converted to XML, again using a Python program.

If no difference list exists, the result of the merging of two layers can be linearised as an XML document straightforwardly. In case the resulting fact base contains a difference list, two different linearizations can be generated. The default processing uses milestone elements to mark the borders of incompatible elements. Alternatively, the technique of fragmentation of elements can be invoked.

4 Conclusion

In this paper it was argued that the problem of representing and processing multiply structured data should be subdivided into two separate problems. First, it is necessary to declare and/or apply to this data elements and attributes defined by different document grammars or belonging to different tag sets. It is desirable to be able to distinguish these elements according to their origins. Furthermore it can happen that the elements belonging to different tag sets mark overlapping regions, which would result in structures that are difficult to handle with SGML-based markup languages. Several proposed solutions for both problems have been discussed. It was argued that the most simple solution, i.e.

annotation of multiple structures or hierarchies in multiple files, can be a way to overcome both problems and that this approach offers many benefits. However, it is necessary to ensure that the multiple files can be represented as a single unit. For doing this, some preconditions have to be accepted by the users of this approach.

5 References

ACH/ACL/ALLC (1994). *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. C. M. Sperberg-McQueen and L. Burnard (eds.). Chicago, Oxford: Text Encoding Initiative.

Barnard, David, Lou Burnard, Jean-Pierre Gaspart, Lynne A. Price, C. M. Sperberg-McQueen, and Giovanni Battista Varile (1995). *Hierarchical Encoding of Text: Technical Problems and SGML Solutions.* In: N. Ide and J. Véronis (eds.). The Text Encoding Initiative: Background and Context, Special Issue of Computers and the Humanities, 29(3), pp. 211-231.

Bayerl, Petra Saskia, Harald Lüngen, Daniela Goecke, Andreas Witt, and Daniel Naber (2003). *Methods for the Semantic Analysis of Document Markup.* In: C. Roisin, E. Munson, and C. Vanoirbeek (eds.). Proceedings of the ACM Symposium on Document Engineering (DocEng 2003), pp. 161-170.

Bray, Tim, Dave Hollander, and Andrew Layman (eds.) (1999). *Namespaces in XML*. W3C Recommendation, World Wide Web Consortium.

Carletta, Jean, Jonathan Kilgour, Tim O'Donnell, Stefan Evert, and Holger Voormann (2003). *The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets.* In:

Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003).

Czmiel, Alexander (2004). *XML for Overlapping Structures (XfOS) using a non XML Data Model.* ALLC/ACH 2004, Joint Conference of the ALLC and ACH, Göteborg.

DeRose, Steve, David Durand, Elli Mylonas, and Allen Renear (1990). *What is Text, Really?* Journal of Computing in Higher Education, 1(2), pp. 3-26.

Durand, David G. (1999). *Palimpest: Change-Oriented Concurrency Control for the Support of Collaborative Applications*. PhD Thesis, Boston University.

Durusau, Patrick and Matthew Brook O'Donnell (2002). *Concurrent Markup for XML Documents*. XML Europe 2002.

Haugen, Odd Einar (2004). *Parallel Views: Multi-level Encoding of Medieval Nordic Primary Sources.* Literary and Linguistic Computing, 19(1), pp. 73-91.

Huitfeldt, Claus and C. M. Sperberg-McQueen (2001). *TexMECS: An Experimental Markup Meta-Language for Complex Documents*. `http://www.hit.uib.no/claus/mlcd/papers/texmecs.html.`

Pianta, Emanuele and Luisa Bentivogli (2004). *Annotating Discontinuous Structures in XML: the Multiword Case.* In: A. Witt, U. Heid, H. S. Thompson, J. Carletta, and P. Wittenburg (eds.). Proceedings of the LREC-Satellite-Workshop on XML-based Richly Annotated Corpora, Lisbon, pp. 30-37.

Renear, Allen, Elli Mylonas, and David Durand (1996). *Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies.* In: International Association for Literary and Linguistic Computing: Selected papers from the ALLC/ACH Conference: Christ Church, Oxford, April 1992. Oxford: Clarendon Press.

Sasaki, Felix and Jens Pönninghaus (2003). *Testing Structural Properties in Textual Data: Beyond Document Grammars.* Literary and Linguistic Computing, 18(1), pp. 89-100.

Sasaki, Felix, Andreas Witt, and Dieter Metzing (2003). *Declarations of Relations, Differences and Transformations between Theory-specific Treebanks: A New Methodology*. In: J. Nivre (ed.). Proceedings of the Second Workshop on Treebanks and Linguistic Theories, Växjö, pp. 141-152.

SGML ISO 8879:1986. *Information processing – Text and office systems – Standard Generalized Markup Language (SGML).*

Sperberg-McQueen, C. M. and Claus Huitfeldt (1999). *Concurrent Document Hierarchies in MECS and SGML*. Literary and Linguistic Computing, 14(1), pp. 29-42.

Sperberg-McQueen, C. M., Claus Huitfeldt, and Allen Renear (2001). *Meaning and Interpretation of Markup*. Markup Languages: Theory & Practice 2(3), pp. 215-234.

Sperberg-McQueen, C. M., David Dubin, Claus Huitfeldt, and Allan Renear (2002). *Drawing Inferences on the Basis of Markup*. In: Proceedings of Extreme Markup Languages 2002.

Tennison, Jeni and Wendell Piez (2002). *The Layered Markup and Annotation Language*. In: Proceedings of Extreme Markup Languages 2002.

Thompson, Henry S. and David McKelvie. *Hyperlink Semantics for Standoff Markup of Read-Only Documents*. In: Proceedings of SGML Europe ’97.

Witt, Andreas, *Meaning and Interpretation of Concurrent Markup.* In: ALLC/ACH 2002, Joint Conference of the ALLC and ACH, Tübingen.

Witt, Andreas, Harald Lüngen, Felix Sasaki, and Daniela Goecke (2004). *Unification of XML Documents with Concurrent Markup.* In: ALLC/ACH 2004, Joint Conference of the ALLC and ACH. Göteborg.

XQuery (2004) *XQuery 1.0: An XML Query Language*. S. Boag, D. Chamberlin, M. F. Fernàndez, D. Florescu, J. Robie, and J. Siméon (eds.). W3C Working Draft, 23 July 2004.

XSL Transformations (1999). *XSL Transformations (XSLT) Version 1.0*. J. Clark (ed.). W3C Recommendation, 16 November 1999.

Andreas Witt
Universität Bielefeld
Fakultät für Linguistik und Literaturwissenschaft
Arbeitsbereich Computerlinguistik und Texttechnologie
Postfach 10 01 31
33501 Bielefeld
Germany
andreas.witt@uni-bielefeld.de
http://coli.lili.uni-bielefeld.de/~andreas/

VP-Fronting in Czech and Polish—A Case Study in Corpus-Oriented Grammar Research *

Roland Meyer

University of Regensburg

Fronting of an infinite VP across a finite main verb—akin to German "VP-topicalization"—can be found also in Czech and Polish. The paper discusses evidence from large corpora for this process and some of its properties, both syntactic and information-structural. Based on this case, criteria for more user-friedly searching and retrieval of corpus data in syntactic research are being developed.

1 Introduction

Word order in Slavic languages is commonly claimed to be free in the sense that many permutations of the words in a sentence are acceptable, given suitable context. While this is most often demonstrated for adjuncts and argument expressions, it holds equally for verbs and verb phrases, cf.[1]

(1) *(Bez względu na to, jaki był cel podjątej*
without regard on that which was aim undertaken-GEN
wyprawy,) prowadzić to będzie do przedłużania
expedition-GEN lead that FUT-AUX to extension
bałkańskiego dramatu.
Balkan-GEN drama

* Thanks are due to the Institute of the Czech National Corpus and to the Institute of Computer Science of the Polish Academy of Sciences which kindly provided the data source for this paper. I also wish to thank Denisa Lenertová and the audiences of the Potsdam Workshop on Heterogenity in Linguistic Databases and of the 13th JungslavistInnen-Treffen for discussion, and the editors for their friendly insistence. All errors are my own.

[1] The glosses are abbreviated as follows: ACC=accusative, AUX=auxiliary, DAT=dative, FUT=future, GEN=genitive, INS=instrumental, MASC=masculine, MP=modal particle, NEG=negation, NOM=nominative, PL=plural, PT=past tense, REFL=reflexive, SBJ=subjunctive, SG=singular.

Interdisciplinary Studies on Information Structure 02 (2005): 87–115
Dipper, S., M. Götze and M. Stede (eds.):
Heterogeneity in Focus: Creating and Using Linguistic Databases

‘(Disregarding what was the goal of the expedition,) it will lead to prolongation of the Balkan drama.’ [Polish]

This type of inversion is reminiscent of the German construction commonly called “VP- topicalization”, as in

(2) a. *Sie hat nicht* [$_{VP}$ *den Peter geküsst*]
she has not the P. kissed
‘She has not kissed Peter.’
b. [$_{VP}$ *den Peter geküsst*]$_j$ *hat sie nicht* t_j
the P. kissed has she not

In both (1) and (2), a non-finite VP or V^0 head has been moved to the left, crossing over the surface position of the governing auxiliary.[2] In (1), only the verbal head has been shifted, leaving behind its directional argument PP, while the whole VP constituent has fronted in (2); but, as (3) and (4) indicate, Polish also allows fronting of a complete VP, and German also allows topicalization of only a partial VP:

(3) *(. . . leży i o Bożym świecie nie wie! . . .) — I jadł kiełbasy*
lies and about god’s world not knows and eat sausage
nie będzie, i gorzałki nie posmakuje?
not FUT-AUX and vodka not tastes
‘(he is lying around and doesn’t know about the world . . .) — He won’t eat sausage, and he won’t try vodka either?’

(4) *Geküsst hat sie den Peter nicht.*
kissed has she the P. not
‘She hasn’t kissed Peter.’ (ex.(2) and (4) after Fanselow (2004))

The two *partial* constituent fronting structures (1) and (4) pose a harder ana-

[2] In (2), the auxiliary is, of course, itself in a derived position, German being a SOV language. In Polish, I assume that the future auxiliary is generated in T^0, from where it has the option of moving up to highfer heads in the extended verbal projection (see Meyer 2004, ch. 4 for explicit argumentation). For the remainder of this paper, it is immaterial whether the auxilary itself moves or not.

lytical problem than the full VP shift illustrated in (2). On one plausible approach, the former actually involve two movement steps, namely (i) the extraction of a subconstituent of the VP—*do przedłużania bałkańskiege dramatu* and *Peter*, respectively—and (ii) the fronting of the rest of the VP, the so-called *remnant* (cf. Müller 1998). This analysis obviously presupposes that the two operations—Scrambling (or extraposition, as argued by Müller 1998) as in step (i) and fronting of a full VP as in step (ii)—exist independently in the language in question, and that their characteristic properties are shared by the partial fronting construction. Fanselow (2004) develops a different approach, in which there is no *remnant* movement. Instead, the subcategorization frames of both the auxiliary and the main verb are merged and their requirements fulfilled on the syntactic level. The moved constituent is not a full VP, but a smaller verbal projection, which does not contain a trace.

The most relevant syntactic properties of VP-fronting in German include the following (cf. Müller 1998, Fanselow 2004):

- A moved VP becomes itself an island for the extraction of one of its subconstituents
- Partial VP-fronting is only possible if the VP targets the SpecC position. Thus, (5-b) is scarcely acceptable in German.

(5) a. *dass [den Peter geküsst] keiner hatte*
that the Peter kissed nobody had
'that nobody had kissed Peter'
b. ?-*dass geküsst keiner den Peter hat

The present paper has two goals: First, it presents the relevant evidence for VP-fronting in Czech and Polish which can be gathered from two large-scale, annotated corpora, namely, the Czech National Corpus (ČNK) and the Corpus of the Institute of Computer Science of the Polish Academy of Sciences (IPI PAN).[3]

[3] See section 4 for information on these corpora. Unless mentioned otherwise, all Czech ex-

Second, it shows *how* this evidence may be accessed and discusses selected design features of these two corpora from the perspective of the user.

2 Clause Structure and Potential Landing Sites in Czech and Polish

There are a few pivotal elements, such as sentential negation, verbal elements, and clitic auxiliaries and pronouns, which can be used to delimit basic clause structure in Czech and Polish. I will briefly present the relevant evidence and a topological overview for orientation.

2.1 Czech

In Czech, sentential negation immediately precedes the main verb, the future auxiliary, and the present or future copula, while it obligatorily follows the clitic past and subjunctive auxiliary, as long as the verb stays below the latter (cf. Junghanns 1999):

(6) a. *To (*ne)jsem/bych (ne)řekl.*
this not-AUX.PT.1SG/AUX.SBJ.1SG not-said
'I didn't/wouldn't say that.'

b. *To vám (ne)budu (*ne)říkat.*
this you-DAT not-AUX-FUT not-say
'This, I won't tell you.'

The most economical way to grasp these positional restrictions is to assume a structure with fixed slots for clitics, negation, the future auxiliary, and the verb phrase, in this order:

XP?	*že* *'that'*	XP?	aux. > refl. > pron. clitics	XP*	NEG	AUX-FUT	XP*	VP

amples in the remainder are from ČNK and all Polish ones from IPI PAN.

Auxiliary and pronominal clitics[4] principally occupy the "second position" of the clause, following the first constituent. In colloquial Czech, they may also occur clause-initially. In embedded clauses, there is an optional slot between the copula and the clitics, which may be filled by an emphasized, focused or topicalized constituent.

However, this is not the only possible structure. The main verb (in the form of the so-called *l*-participle), including negation, may precede the clitics. This can only happen in matrix clauses, where there is no first constituent XP:

(7) *(Ne)řekl* *(*ne)bych* *to.*
not-said-MASC not-AUX-SBJ.1SG this
'I wouldn't say that.'

The movement of the participle differs fundamentally from German VP-topicalization in that it cannot take along any further material (8); therefore, it has been argued to involve V^0 head movement rather than phrasal movement.

(8) a. **Posílal dopisy jsem* *ti* *pravidelně každý týden.*
sent letters AUX-PT.1SG you regularly every week
(Avgustinova & Oliva 1997, 40)

b. **... že* *nedal* *by* *mu* *to.*
that not-gave SBJ him this
(Veselovská 1995, 149)

Since participle movement is so restricted in Czech, I will concentrate on the movement of *infinitival* VPs—including partial VPs—to the pre-clitic position, which *is* comparable to German VP-topicalization. However—as we will see below—there is a further landing site for this VP-movement between the clitics and the negation slot.[5] I will refer to the former as "high VP-fronting", and to the latter as "low VP-fronting" in the remainder.

[4] These include the past and subjunctive auxiliary, as well as the short forms of pronouns.

[5] Cf. (5-b) above, which shows that *partial* VP-Scrambling to the left edge of the middle field is excluded in German.

A set of examples for VP-fronting to the left of the clitics is mentioned by Avgustinova and Oliva (1997, 40), including infinitives as in (9-a) and passive participles as in (9-b):[6]

(9) a. *Posílat dopisy ti budu pravidelně každý týden.*
send letters you FUT-AUX regularly every week
'Send letters to you I shall regularly every week.'
b. *Srdečně uvítáni domorodým obyvatelstvem jsme*
cordially greeted original-INS inhabitants-INS AUX-PT.1PL
rozhodně nebyli.
certainly not-been
'For sure, we were not greeted by the original inhabitants cordially.'

2.2 Polish

There are less obvious structural markers in the Polish clause than in the Czech one. Clitic pronouns (so-called weak forms), past tense verbal person and number affixes and the subjunctive marker *by* do not obey a strict second-position requirement:

(10) *Do której kategorii pan by się zaliczył?*
in which category sir SBJ REFL counted
'Into which category would you put yourself?' (APTC)[7]

However, there are restrictions on the relative linear order among these clitic elements (cf. Witkoś 1996, 165):

(11) a. *Maria (*go*) spotkała (*go*) w środę.*
M.-NOM him met him on Wednesday-ACC

[6] The authors refer to these as "partial VP-fronting"; actually, they rather involve full VP-fronting (maybe except for the clitic raising in (9-a)). However, truly *partial* VP-fronting is also possible in Czech (see below for examples).

[7] This example stems from Adam Przepiórkowski's "Toy Corpus", an early predecessor of the IPI PAN corpus, which has been disconnected recently.

'Mary met him on Wednesday.'

b. *Maria* by *(*go*) (/*go by*) spotkała (*go) w środę.*
M.-NOM SBJ him him SBJ met him on Wednesday-ACC
'Mary would have met him on Wednesday.'

c. *Maria (*go) spotkałaby (go) w środę.*
M.-NOM him met-SBJ him on Wednesday-ACC

The pattern in (11) is commonly accounted for via two assumptions (Witkoś 1996, Błaszczak 2001): (i) the subjunctive marker is generated above the pronominals and none of them move, and (ii) the main verb may move up to the position of the subjunctive marker. The behaviour of the verbal person and number (PN-) affixes is similar, but not identical, cf. Dornisch (1998) and Błaszczak (2001):

(12) a. *My*śmy *(*go*) (/*gośmy*) widzieli (*go) wczoraj.*
we-PT.1PL him him-PT.1PL saw him yesterday
'We saw him yesterday.'

b. *My (*go*) widzieli*śmy *(*go*) wczoraj.*
we him saw-PT.1PL him yesterday
'Wir sahen ihn gestern.'

Since the past PN-marker can follow the main verb even if the clitic pronominal stays above it (12-b), it seems that it may occupy either of two positions: a high one above the pronominal clitics (but below the subjunctive marker), and a low one next to V^0.[8]

The relative order of verbal elements and negation supports this view: sentential negation follows the subjunctive and past tense PN-markers if they occur in their high position, but it precedes the main verb or the future auxiliary.[9] These considerations lead to the following topological picture of the Polish

[8] In the latter case, the verb has to stay low, because otherwise the excluded sequence [*go* ... *-śmy* ... V] would be predicted again (cf. (12-b)).

[9] If the main verb raises up to the position of the subjunctive or PN-marker as in (11-c), it takes the negation along, resulting in the order Neg+V+*go*.

clause:

że *'that'*	XP*	(verb+) PN- / SBJ-marker	pron. clitics	XP*	NEG	FUT-AUX	XP*	VP (+ PN-marker)

Note that—exactly as in Czech (cf. (8))—no VP, complete or partial, may raise to the slot of the first XP* in this schema:

(13) *[*Poszli do szkoły*]$_k$ *-śmy* t_k
went to school PT.1PL
(Bański 2001, 185)

To be sure, a detailed corpus search using the query in (14-a) yielded at least one example of this kind:[10]

(14) a. `[pos=praet] [] "by" [pos=aglt] within s`
b. *. . . , pochował ja bym go tak, żeby go i na sąd*
hidden I SBJ-1SG him so that him also on court
ostateczny nie znaleźli.
last not found-3PL
'. . . I would hide him so that he would not even be found on judgment day.' (Sienkiewicz 1895, IPI PAN)

While (14-b) may be doubtful (coming from an earlier stage of modern Polish, colloquial in style), there is no problem in the standard language with an XP and the verb raising independently, as in

(15) *Co radziłbyś bliskiemu sobie młodemu człowiekowi,*
what advise-SBJ-2SG close-DAT REFL-DAT young-DAT person-DAT
aby zrobił po ukończeniu szkoły?
that did after finish school
'What would you advise a young person close to you to do after finishing school?'

[10] The above query would read "sequence of a past tense form, an arbitrary token, *by*, and a PN-marker within one clause" in natural language (cf. Przepiórkowski 2004). The syntax is obviously similar to the one of the CQP query language (Christ, 1994).

I conclude that participles generally raise as heads, not as VPs, in modern Polish (like in Czech), but the motivation for their movement has to be completely independent of the requirement to support the clitics (other than in Czech). Relevant constructions for the purposes of this paper mainly include infinitival VPs raised to the position before the future auxiliary ("low VP-fronting" in the remainder) or to the left of the preverbal PN- and subjunctive marker ("high VP-fronting").

3 Results of the Corpus Query

In this section, I will show some results of a corpus-oriented investigation of infinitival VP-fronting constructions in Czech and Polish, based on the Czech National Corpus (Český Národní Korpus, ČNK), and the IPI PAN corpus of Polish (IPI PAN, Przepiórkowski 2004). Both corpora have been lemmatized and annotated for morphosyntactic categories using a stochastic tagger. Nevertheless, there are important differences in the design of the annotation (see section 4). I use corpus evidence in a purely qualitative manner here, as an indication of what constructions can be found with some basic frequency in the two corpora, and what contexts they occur in. Needless to say, something which is not in a corpus, however large it may be, can still be part of the language and its grammar. But we can at least challenge restrictive intuitive judgments by counterevidence from the corpus, or support an intuitive restriction by the lack of the latter. Given that the VP-fronting data are very context-dependent and not always easy to judge for informants, this is already of some help.

3.1 Czech

3.1.1 High VP-fronting

High VP-fronting of an infinitive in Czech may easily be searched in the ČNK using an expression like[11]

(16)
```
[tag="Vf."] [tag!="Z.*" & word!="a"]*
"((js[iemt][me]?)|(sis)|(ses))" within s
```

We find that the infinitive may target the slot between the complementizer and the clitic auxiliaries (17), a configuration which is known to be disallowed with participle raising (8-b):

(17) *Samozřejmě uznávám, že ohánět jsem se po něm*
certainly acknowledge-1.SG that beat AUX-1SG REFL for him
neměl.
not-should
'Of course I acknowledge that I shouldn't have beaten him up.'

Second, there are many cases of a *complete* VP being fronted across the clitics, as in Avgustinova & Oliva's (1997) examples mentioned above:

(18) [*zahodit ji a vydat se pěšky na útěk k východním*
throw-away her and start-out REFL on-foot on flight to Eastern
hranicím směr domů]$_k$ *jsi nemohl* t_k
borders direction home AUX-PT.2SG not-could
'You couldn't throw her away and start out on foot, fleeing towards the Eastern borders.'

Third, we also encounter some clear cases of *partial* VP-fronting, as in

(19) *Usadit nastálo jsem se chtěl v Patagonii*
settle-down constantly AUX-PT.1SG REFL wanted in P.

[11] In ordinary language, "a sequence of an infinitive, any number of non-punctuation and non-*a* '*and*', and a past auxiliary, within one sentence".

'I wanted to settle down constantly in Patagonia.'

Interestingly, the search hits show a linear order restriction such that the infinitival verb always *precedes* its objects. The only cases in which another, sentence-initial word preceded the infinitive consisted of stacked infinitives, as in

(20) [$_{VP}$ *Pomoci objasnit celý případ*] *by mohli taxikáři, kteří*
help explain whole case SBJ could taxi-drivers who
napadení turistů viděli.
attack tourists-GEN.PL saw
'The taxi drivers, who saw the attack on the tourists, could help to clarify the whole case.'

Intuitive judgments support this impression, cf.

(21) **celý případ objasnit by mohli taxikáři*
whole case explain SBJ could taxi-drivers

Obviously, the base order within VP (this time, verb final) has to be preserved also in analogous German examples:

(22) a. *Den ganzen Fall aufklären könnten die Taxifahrer, ...*
the whole case clarify could the taxi-drivers
'The taxi drivers could clarify the whole case ...'
b. **Aufklären den ganzen Fall könnten die Taxifahrer, ...*
clarify the whole case could the taxi-drivers

An explanation for this pattern could build on the idea that in the ungrammatical (22-b) and (21), two constituents move independently, while there is only one landing site available. Under the view of scrambling as A-movement to some specifier in the Agr- or T-domain (Zybatow and Junghanns, 1998), the derivation of (21) would have to involve an intermediate step in which the object and the main verb do not form a constitutent any more, so they cannot move as one VP.

As concerns information structuring, the corpus examples display a pattern

of contrastive topic plus sentence-final focus throughout:

(23) *(O kolej žádá asi šestnáct set lidí.)* [$_{\text{TOP}}$
for dormitory ask probably sixteen hundred people
Nabídnout] jsme mohli pouze [$_{\text{FOC}}$ *pět set sedmdesát*
offer AUX-PT.1PL could only five hundred seventy
míst, která uvolnili letošní absolventi.]
places which freed this-year's alumni
'(About 1600 persons apply for the dormitory.) We could offer only 570 places, (which opened after this year's alumni left.)'

3.1.2 Low VP-fronting

VP-fronting to a position between the clitics and the finite verb was searched using a query like[12]

(24)
```
"((js[iemt][me]?)|(sis)|(ses))" [tag!="Z.*" &
lemma!="a"]* [tag="Vf.*"] [tag!="Z.*" &
lemma!="a"]+ [tag="Vp."] within s
```

The example in (25) illustrates the fronting of a complete VP to this area, with the base order *verb*>*object* preserved:

(25) *(… pak je to nejen proto,) že jsem* [$_{\text{VP}}$ *udělat kariéru*
then is this not-only because that AUX-PT.1SG make career
] *chtěla a chci, ale také proto, že …*
wanted and want-1SG but also because that
'(…then this is the case not only) because I wanted and want to make a career, (but also because …)'

The low landing site is below the subject position, as (26) indicates:

[12] In natural language, "a sequence of a clitic auxiliary, any number of non-punctuation and non-*a* 'and', an infinitive, at least one token which is not punctuation and not *a* 'and', and a finite verb, within one sentence."

(26) *Pokud by zákonodárci schválit misi nestihli, ...*
if SBJ legislators approve mission not-managed
'If the legislators would not manage to approve of the mission ...'

The moved VP can be partial—e. g., if one of its subconstituents goes into topic position:

(27) *Zadarmo jsem se jí [vzdát]$_l$ ale nechtěla t_l.*
for-free AUX-PT.1SG REFL her give-up but not-wanted
'But I didn't want to give her up for free.'

As opposed to the pre-clitic, high landing site, the low landing site imposes no linear restrictions on the fronted VP—e. g., the object can precede its governing infinite verb:

(28) *... ale dál jsem se politice* [*věnovat*]$_k$ *skutečně*
but further AUX-PT.1SG REFL politics devote really
nechtěl t_k.
not-wanted
'...but I really did not want to devote myself to politics any further.'

This is not very surprising, given that scrambling to the front of the VP is iterable in Czech anyway, so that there are always enough structural positions available to front a VP and one of its elements independently. As far as information structural distinctions are concerned, low VP-fronting seems to serve mainly one purpose: to move background material out of the focus domain, which then consists only of the right-peripheral—mostly negated—finite verb. All of the 106 relevant examples extracted from the ČNK conformed to this pattern.

(29) *"...Za druhou půli jsme však* [$_{VP}$ *vyhrát]$_i$ určitě* [$_{FOC}$
during second half AUX-1PL MP win certainly
zasloužili t_i]," řekl trenér
deserved said coach
' "But in the second half we surely deserved to win", said the coach.'

3.2 Polish

3.2.1 High VP-fronting

The IPI PAN query for Polish VP-fronting illustrates some pecularities of the annotation scheme used by Przepiórkowski (2004):[13]

(30)
```
[pos=inf] [pos!=interp & pos!=praet &
base!="((i)|(lub)|(albo))"]* "by" [pos=aglt]
within s
```

Some common word classes are being replaced by more fine-grained distinctions (`inf, praet`), which correspond more closely to differences in inflectional type. Furthermore, the PN-marker is always analyzed as a seperate token (of word class `aglt`), even if it occurs immediately after the main verb. These changes result from a careful and linguistically motivated tagset design (cf. Przepiórkowski and Woliński 2003): As mentioned above, the subjunctive marker and the PN-marker, although intuitively part of the verbal inflectional paradigm, may attach to various constituents phonetically, and also orthographically. It is therefore easiest to treat these items as tokens of their own, ignoring orthographic word boundaries at the level of morphosyntax (cf. section 4 for further remarks on the tagset).

The query in (30) mainly uses the subjunctive marker to delimit the high landing site of fronted infinite VPs. Since the analysis involving raising of the finite verb up to the subjunctive marker may be controversial, I will rely on subjunctive markers occurring separately as topological markers. Fronted VPs may be either complete (31) or partial (32):

[13] The query roughly reads as “a sequence of an infinitive, an arbitrary number of tokens which are neither punctuation nor a past tense verb nor one of *i* ‘and’, *lub* ‘or (incl.)’ *albo* ‘or (excl.)’, *by*, and an agglutinative affix (*-śmy, -ście*, etc.), all within one sentence”.

(31) *W każdym bądź razie bodaj na części zasobów muzealnych*
in every possible case MP on part stock-GEN.PL museum
uwłaszczyć byśmy się mogli, . . .
acquire SBJ-AUX-1PL REFL could
'But every single time we could have acquired at least part of the museum stocks, . . . '

(32) *Żyć bym bez nich nie potrafił.*
live SBJ-AUX-1SG without them not managed
'I would not have managed to live without them.'

As (31) shows, there is no constraint on linear order within the fronted VP, as opposed to the analogous case in Czech. In fact, more than one constituent may precede the fronted VP, cf.

(33) *. . . , ale ja niczego absolutnie wykluczać bym w tej materii*
but I nothing absolutely exclude SBJ-AUX-1SG in that issue
nie chciał.
not wanted
'. . . but I would not want to exclude anything in this matter.'

In the light of (33), it is not surprising that several subconstituents of the infinite VP may move independently to several stacked specifiers or adjunction positions, resulting in a linear order as in (31).

Regarding information structure, in the corpus examples either the whole fronted VP (34) or at least a subconstituent of it (35) functions as a contrastive topic:

(34) *. . . bez sieci nie potrafię już żyć, a pewnie* [$_{\text{TOP}}$ *umrzeć*] *też*
without net not be-able already live and surely die also
bym nie potrafił.
SBJ-AUX-1SG not be-able
'. . . without a net(work) I cannot live any more, and probably also could not die (without it).'

(35) *(Ponieważ jednak los tej ustawy w chwili obecnej jest wysoce*
because however fate this law-GEN in time present is highly
hipotetyczny,) dalej w tej kwestii posunać bym się
hypothetical further in this question move SBJ-AUX-1SG REFL
nie mógl.
not could
'(But since the fate of this law is highly hypothetical at present,) I could not go any further in this issue.'

3.2.2 Low VP-fronting

The query for VP-fronting to the lower landing site looks very similar to (30), except for the infinite VP *following* the subjunctive and PN-marker. A partial VP-fronting example from the IPI PAN corpus would be

(36) *. . . , ale też długo bym leżeć nie chciała, (bo*
but also long SBJ-AUX-1SG lie not wanted because
bym chyba nie wytrzymała.)
SBJ-AUX-1SG probably not stand
'. . . but I wouldn't want to lie around for long, (because I probably couldn't stand it).'

As in Czech, the fronted VP mostly moves out of the domain of focus, leaving behind a minimally focused main verb. However, there are also cases in which the most plausible focus would be on the whole finite VP or even on the whole clause:

(37) *("Miasto Lozanna zresztą dość nudne. . . . byłoby nam tu*
city L. by-the-way rather boring SBJ-3SG us here
dobrze,) gdybyśmy przywyknąć mogli do cudzej ziemi!"
good if-SBJ-AUX-1PL adjust could to foreign country
'("By the way, the city of Lausanne is rather boring . . . we would feel good here,) if we could adjust to the foreign country." '

The construction also occurs with enumerations of non-minimal focus domains,

as in (3) above. At least in these two cases, there is no obvious information structural side effect of VP-fronting in Polish.

4 Corpora of Slavic Languages as a Source for Syntactic Research

In this section, I will take a step back and consider the usability of the available Czech and Polish corpora for research into syntax and information structure of the kind presented in the preceding section. The intention is not a confrontative comparison of the ČNK and the IPI PAN corpus, but rather a collection of ideas for more user-friendliness and search power.

4.1 Corpus Annotation (POS-Tagging and Morphosyntactic Analysis)

The two main features we used in the queries were regular expressions over word forms and part-of-speech (POS) tags. The query language of the ČNK is essentially the CQP language (cf. Christ 1994); the query language of the IPI PAN corpus is very similar, but offers some interesting modifications (see below). Both corpora are fully lemmatized.

The ČNK (or rather, its publicly accessible part SYN 2000) consists of about 100 million tokens, with about 60 % taken from the public press, 20 % from non-fictional literature, and 15 % from fictional literature. The mixture of texts has been carefully considered, and SYN 2000 nowadays functions as a stable reference corpus. Around this core, two larger corpora of spoken Czech and several other collections have been made available.

Rather than mere POS-tagging, a full morphosyntactic analysis was conducted on SYN 2000, using a set of more than 2000 different theoretically possible tags. Accuracy of the stochastic tagging was estimated at about 93 % in 1998 (Hajič and Hladká 2000). The remaining errors, partly accidental, partly systematic, have evoked harsh criticism by members of the Institute of the Czech National Corpus and their colleagues. Efforts are being made to improve on the

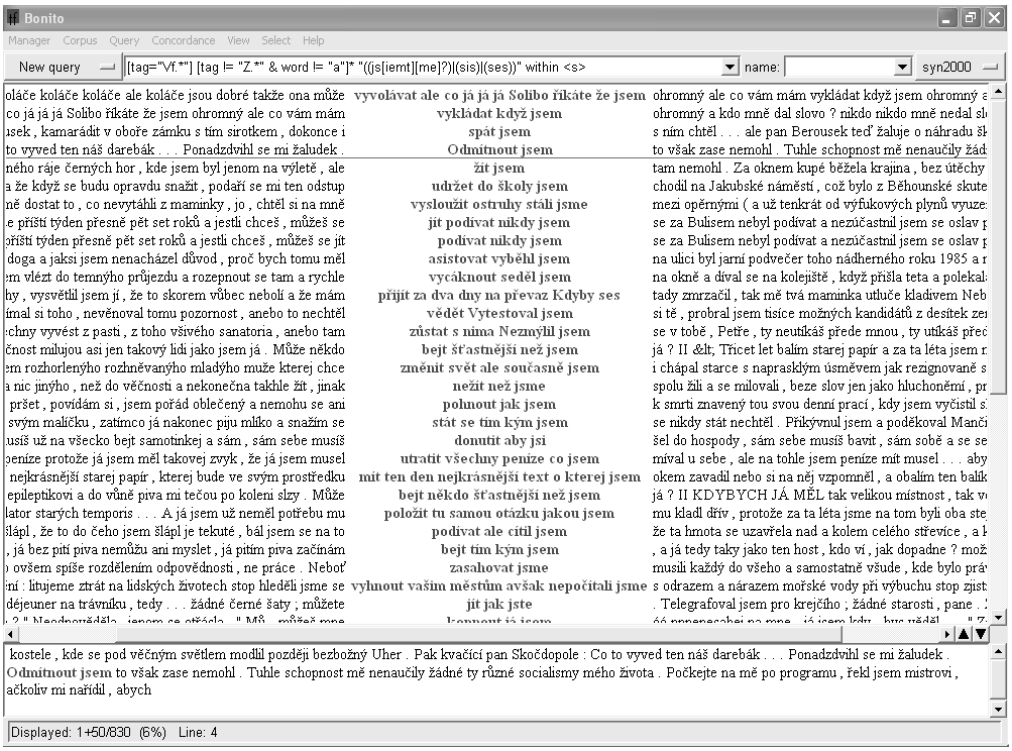


Figure 1: The Czech National Corpus and its query interface Bonito

tagging by using a mixture of stochastic and rule-based methods ((Hajič et al., 2001)). The ČNK, just as the IPI PAN corpus or the MULTEXT-East corpora (Erjavec and Monachini 1997), uses a positional tagset, i. e., the value of each grammatical category is encoded by one character in a fixed position in the string which makes up a tag.

The IPI PAN corpus was released by the Institute of Computer Science of the Polish Academy of Sciences as a first version in June 2004. It currently consists of about 70 million tokens, or rather “segments” (Przepiórkowski 2004): the clitic PN- and subjunctive marker are regarded as units of their own right and are written seperately from their prosodic hosts. Half of the materials contained in the corpus are newspaper texts, 20% are fictional (prose), about 10% scientific writing, 5% law texts, and 15% session protocols from the Polish parliament. As Przepiórkowski (2004) states himself, this collection is rather opportunistic

than balanced and will have to be improved upon at a later stage.

Figure 2: The IPI PAN Corpus and its query interface Poliqarp

The POS- and morphosyntactic tagging was done by stochastic methods; however, at least two new ideas set this corpus apart from other stochastically tagged sources like the British or the Czech National Corpus:

First, the set of POS tags has been designed in a somewhat non-traditional way which relies as little as possible on lexical semantic groups, and thus could make word class recognition easier. Tokens were grouped into different word classes whenever they differed systematically with respect to their sets of possible grammatical categories (i. e., when they belonged to different *flexemes* in the terminology of Przepiórkowski and Woliński (2003), originally attributed to Janusz Bień.). Thus, instead of a mixed bag like “adjective”, there are the classes *adjective*, *ad-adjectival adjective* (e. g., *‘polsko-niemiecki’ Polish-German*), and *post-prepositional adjective* (e. g., *(po) polsku ‘(in) Polish’*), of which only the

first one shows a complete inflectional paradigm (Przepiórkowski 2004, 28). The number of possible tags is reduced dramatically by disregarding all the lexical semantic sub-distinctions of pronominals (interrogative, relative, demonstrative, etc.), grouping them with adjectives or nouns as far as possible, according to their inflectional behaviour. Most of the lexical semantic distinctions can be regained anyway by doing a lemma search; and there is virtually no hope that any automatic tagger will be able to detect the correct tags for homonymous pronouns belonging to different semantic classes. To give an example, a search of interrogative vs. (free) relative pronouns in the ČNK yields words from both categories without any obvious pattern.[14] A similar rationale as in the case of IPI PAN seems to have played a role in the design of the Stuttgart-Tübingen Tagset for German. Unfortunately, success rates of the IPI PAN tagging are not yet avaliable.

Second, the set of possible morphosyntactic tags before stochastic disambiguation is retained and may be searched. This can be useful and sometimes superior to a forced disambiguation. To give an example, a search was conducted for discontinuous NPs in dative case, which are split into the attributive adjective and the seperate head noun (a so-called *Left-Branch Extraction*). Since the dative case can be homonymous to other cases, depending on inflectional class, a forced stochastic disambiguation—as in most present-day corpora, including the ČNK—will inevitably yield errors:

(38) *ke každé otázce z přízemí*
to every-DAT question-DAT from stalls-*DAT/$\surd$GEN

In this example from ČNK, the genitive on *přízemí* has been wrongly tagged as a (homonymous) dative. All these examples will have to be reconsidered

[14] This is not very surprising, given the huge number of homonyms: In the case of Russian, a traditional, semantically based classification pronouns, as it was considered for some time for the corpora of the Tübingen Sonderforschungsbereich 441, would lead to about 600 different morphosyntactic tags only for this category, which is almost half the size of the whole tagset.

"manually" by the user, in order to filter out the true datives. With the ambiguous tagging of IPI PAN, however, the query may be limited to only those hits in which initial morphological analysis has already yielded a single, *non-ambiguous* form—disregarding all the potentially wrong tags.

Figure 3: Searching for non-ambiguous forms in the IPI PAN corpus

From a user's point of view, keeping track of ambiguous and unambiguous tags is thus certainly an advantage. At the same time, however, the need for better success rates of automatic disambiguation is obvious and has been acknowledged repeatedly by the corpus designers.

4.2 Comfortable Querying

Given the range of corpus users (a part of the ČNK, SYNEK, is even being propagated in secondary schools), user friendliness has become an increas-

ingly important issue.[15] The ČNK has been distributed together with a comfortable and easily understandable search tool from the beginning; this tool, called G(raphical)CQP, which was programmed by Pavel Rychlý of the NLP laboratory of Brno University, has been developed further into the corpus server *Manatee* and the client viewer *Bonito*, offering even more possibilities. A purely web-browser based version has been announced for 2005. To name just a few of the many interesting features of Bonito, there is (i) a graphical representation and construction of queries; (ii) Bonito offers a stepwise refinement of the queries by imposing positive and negative filters on search results. Nicely, the user can always return to a previous set of hits in case a filter did not give the intended result. (iii) Bonito contains enhanced statistic functionality for research into collocations, (iv) handling of user-defined subcorpora, and (v) export functions for search results. These possibilities greatly increase accessibility and would definitely be desirable for the corpora of other languages as well. Compared to the search tools of the British National Corpus, the recently initiated Russian National Corpus, and the German COSMAS corpus, the search tool of the Czech National Corpus definitely sets new standards.

Neither of (i)-(iv) have yet been realized in Poliqarp, the first version of the search tool coming with the IPI PAN corpus. However, at least one idea which increases searching comfort in Poliqarp deserves to be mentioned: the inclusion of tag aliases. E. g., instead of learning that the fourth position of each tag encodes number and the fifth position case, thinking up a regular expression over tags, and then typing in `[tag="...P3.*"]`, the user may simply state `[case=dat & num=pl]`. Given the large number of theoretically possible tags and their sometimes counterintuitive positional encoding (e. g. `[tag="Vf.*"]` for verbs in the *infinitive* in ČNK), this is extremely helpful. It would be even nicer if the user had the alternative to input the values of gram-

[15] The ČNK and the IPI PAN textual source data is not handed over to the user, so (s)he has to rely on the search tools provided by the corpus designers.

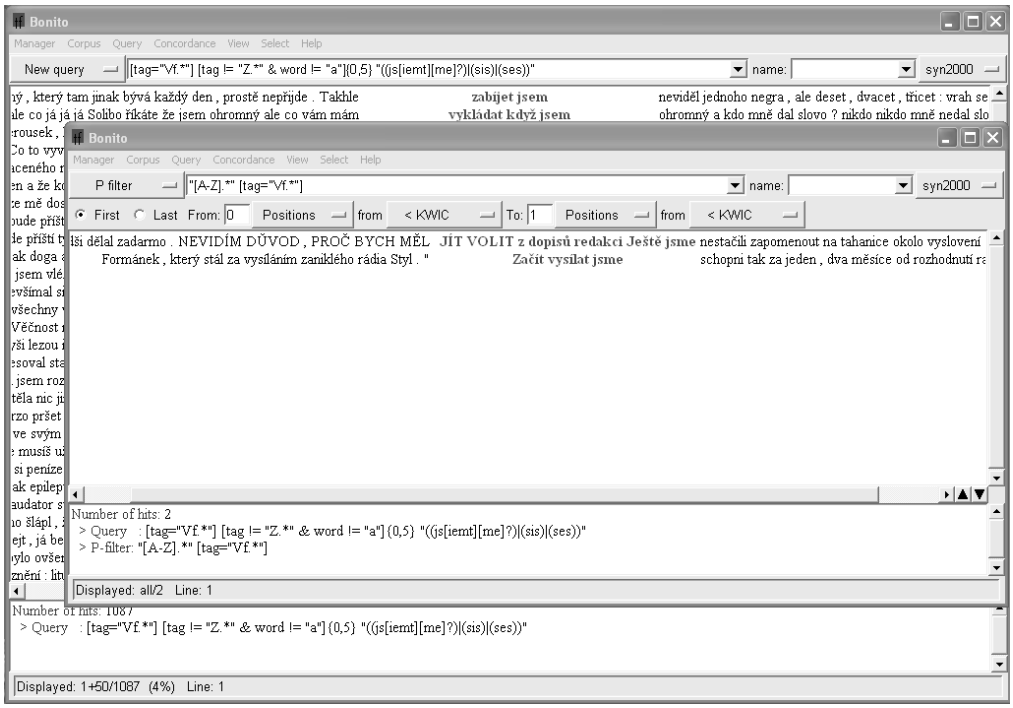


Figure 4: Stepwise filtering of query results in Bonito

matical categories by selecting from menues or simply checking boxes (as in the new Russian National Corpus).

4.3 A Parsed Corpus for Czech: the Prague Dependency Treebank

Both VP-fronting constructions involve displacement from an assumed base position to the left; in partial VP-fronting, even more cases of discontinuous constituency occur. We have seen above that instances of these constructions can be found by relying on morphosyntactic tags and some post-editing and filtering by hand. However, it would be more comfortable to run a search on discontinuous syntactic constituents directly. Queries over syntactic structures can be conducted for Czech using the Prague Dependency Treebank (PDT), a pseudo-random selection of about 55 000 authentic sentences from the ČNK which have been hand-annotated according to the theory of Functional-Generative Descrip-

tion (Sgall et al. 1986). This corpus may be searched on-line, using the search tool Netgraph (developers: R. Ondruška and J. Mírovský). The syntactic structures in the PDT are trees which encode dependency relations and relative linear order between words. A sample query for discontinuous VP constituents and one of its hits is the following:

(39) a.
```
[]([tag=Vf*,ord=1]([ord>=3]),
[afun=AuxV,ord=2])
```

b. *Pomoci by mu v tom měli i noví hráči.*
help AUX-SBJ him in this should also new players
'Also the new players should help him with this.'

The values of the feature `ord` in this case encode that the VP is supposed to be split, i. e., the query concerns an infinitive (first word) with an auxiliary verb as its sister (second word) and a daughter of the infinitive as third or later word.

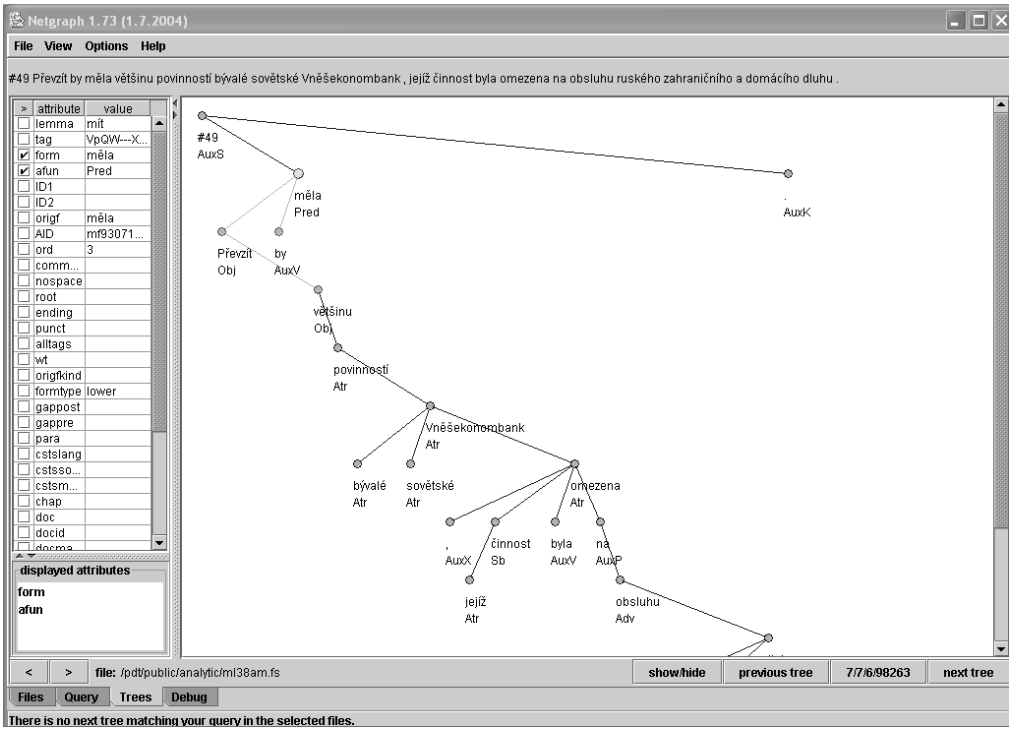


Figure 5: Discontinuous VP-fronting in the Prague Dependency Treebank

A second version of the treebank, including annotation for topic, focus, and contrast (in the sense of Hajičová et al. 2000), has been announced for 2005. It may then serve as a starting point for studies of information structure, but the user still has to go back to the original ČNK and perform a second search to see the context of a retrieved sentence; the search tools of PDT and ČNK are not integrated at present. Although the PDT is definitely valuable for a large scale of natural language processing tasks, it seems that empirical linguistic studies will still have to rely on searching truly large corpora by lower-level annotation such as lemmata and morphosyntactic tags, as well as by forms and collocations.

4.4 Further Freely Accessible Corpora of Slavic Languages

Many modern Slavic languages can already be investigated on the basis of large corpora with free online access. Among those offering at least a search by regular expressions over word forms are the Oslo Corpus of Bosnian Texts, the Croatian National Corpus, the Serbian National Corpus, the Russian Corpora of the Sonderforschungsbereich 441 (University of Tübingen), the Slovak National Corpus, and the Polish Corpus of the PWN publishing house. POS annotation has been provided for the Czech National Corpus, the Polish IPI PAN Corpus, a small portion of the Tübingen Russian Corpora, and, increasingly, for the Russian National Corpus. The MULTEXT-East project has produced POS-annotated versions of Orwell's "1984" for Czech, Slovenian, and Bulgarian. Syntactically analyzed corpora are currently available for Bulgarian (HPSG treebank) and Czech (Prague Dependency Treebank).[16] Work on *diachronic* corpora of Slavic languages has been started at Charles University, Prague (Old Church Slavonic and Czech), the University of Sofia (Bulgarian), and the University of Regensburg (Russian). This short overview is necessar-

[16] IPI PAN has developed a "test suite" of HPSG-parsed Polish sentences, which is, however, rather intended as an overview of possible sentence structures than as a corpus of natural language.

ily incomplete, updated collections and links to web pages may be found e. g. at `www.uni-tuebingen.de/uni/nss/docs/Korpora.html` and at `www-slavistik.uni-regensburg.de/Corpus`.

5 Conclusion

Corpus evidence from ČNK and IPI PAN indicates that in Czech, high VP-fronting is the non-iterable movement of a single—complete or partial—constituent, which does not allow for internal word order variation. In the other three cases considered, i. e. in Czech low VP-fronting and in Polish in general, there are no such restrictions: VP-movement can be accompanied by further fronting operations to the same area, and VP-internal word order is basically free. Theoretically, high VP-fronting in Czech should thus be analyzed as movement to an A'-specifier (SpecC), while the other three operations end up in an adjunction position—either above T (Polish high VP-fronting) or between T and the main VP (Polish and Czech low VP-fronting). In terms of contextual conditions and information structure, high VP-fronting in our Czech examples always involves a contrastive topic, while low VP-fronting is movement of background material to the left; both combine with a minimimal, right-peripheral focus. In Polish, the high fronting also favors an interpretation as a contrastive topic, while low VP-scrambling seems to be compatible with various information structural partitions of the sentence.

To arrive at these generalizations, we made use of corpus data in the following ways: (i) Single intuitive judgments were supported (i. e., not falsified) by a large body of original data. This holds for most of the evidence on clitic placement in section 2. (ii) Naturally occurring contexts were evaluated, resulting in statements about their relative frequency. This is the case for the information structural effects of VP-fronting reported above. (iii) Of two theoretically possible constructional variants, one could be found with a certain basic frequency,

but the other did not occur at all. The corpus thus indicates a candidate for a restriction, which has to be checked against intuitive judgments. This was done, e. g., for the linear order restriction with Czech high VP-fronting. Technically, we simply searched by sequences of regular expressions over words, lemmata, and morphosyntactic descriptions. The user-friendliness for this process could be enhanced considerably if the following features, partly realized already in ČNK and IPI PAN, became standard: (i) retrieval of (un)ambiguous tags before automatic disambiguation; (ii) stepwise refinement of search hits, handling of user-defined subcorpora, easy export of hits; (iii) statistic functions for research into collocations; (iv) aliases or a graphic interface for the input of tags.

Bibliography

T. Avgustinova and K. Oliva. On the Nature of the Wackernagel Position in Czech. In U. Junghanns and G. Zybatow, editors, *Formale Slavistik*, volume 7 of *Leipziger Schriften Zur Kultur-, Literatur-, Sprach- und Übersetzungswissenschaft*, pages 25–57. Vervuert Verlag, Frankfurt/Main, 1997.

P. Bański. Last Resort Prosodic Support in Polish. In G. Zybatow, U. Junghanns, G. Mehlhorn, and L. Szucsich, editors, *Current Issues in Formal Slavic Linguistics*, volume 5 of *Linguistik International*, pages 179–186. Peter Lang, Frankfurt, 2001.

J. Błaszczak. *Investigation into the Interaction between the Indefinites and Negation*, volume 51 of *studia grammatica*. Akademie-Verlag, Berlin, 2001.

O. Christ. A modular and flexible architecture for an integrated corpus query system. In *COMPLEX'94*. Budapest, 1994.

E. Dornisch. Auxiliaries and Functional Projections in Polish. In Ž. Bošković, S. Franks, and W. Snyder, editors, *Annual Workshop on Formal Approaches to Slavic Linguistics: The Connecticut Meeting, 1997*, volume 43 of *Michigan Slavic Materials*, pages 183–209. Michigan Slavic Publications, Ann Arbor, 1998.

T. Erjavec and M. Monachini. *Specifications and Notation for Lexical Encoding (= COP Project 106 MULTEXT-East Final Report)*, 1997. http://nl.ijs.si/ME/CD/docs/mte-d11f/.

G. Fanselow. Against Remnant VP-Movement. ms., Universität Potsdam, 2004.

J. Hajič, P. Krbec, P. Květoň, and Vl. Petkevič. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 260–267, Toulouse, 2001. Morgan Kaufman Publishers.

E. Hajičová, J. Panevová, and P. Sgall. A manual for tectogrammatic tagging of the prague dependency treebank. Technical report, Institute of Formal and Applied Linguistics / Center for Computational Linguistics, Prague, 2000.

J. Hajič and B. Hladká. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *COLING-ACL'98*, pages 483–490. Morgan Kaufman, San Francisco, 2000.

U. Junghanns. Generative Beschreibung periphrastischer Konstruktionen des Tschechischen. In T. Anstatt, R. Meyer, and E. Seitz, editors, *Linguistische Beiträge zur Slavistik aus Deutschland und Österreich. VII. JungslavisInnen-Treffen, Tübingen/Blaubeuren 1998*, pages 133–165. Sagner, München, 1999.

R. Meyer. *Syntax der Ergänzungsfrage. Empirische Untersuchungen am Russischen, Polnischen und Tschechischen*, volume 436 of *Slavistische Beiträge*. Otto Sagner Verlag, München, 2004.

G. Müller. *Incomplete Category Fronting*. Kluwer, Dordrecht, 1998.

A. Przepiórkowski. *The IPIPAN Corpus. Preliminary Version*. Institute of Computer Science, Polish Academy of Sciences, Warszawa, 2004.

A. Przepiórkowski and M. Woliński. A morphosyntactic tagset for polish. In P. Kosta, J. Błaszczak, J. Frasek, L. Geist, and M. Żygis, editors, *Investigations into Formal Slavic Linguistics. Contributions of the Fourth European Conference on Formal Description of Slavic Languages – FDSL IV*, volume 1, pages 349–362. Peter Lang, Frankfurt, 2003.

P. Sgall, E. Hajičová, and J. Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Academia, Praha, 1986.

L. Veselovská. *Phrasal movement and X^0-morphology: word order parallels in Czech and English nominal and verbal projections*. PhD dissertation, Univerzita Palackého, Olomouc, 1995.

J. Witkoś. Pronominal Argument Placement in Polish. *Wiener Linguistische Gazette*, 57-59:147–194, 1996.

G. Zybatow and U. Junghanns. Topiks im Russischen. *Sprache und Pragmatik*, 47:1–57, 1998.

Roland Meyer
Universität Regensburg
Institut für Slavistik
Universitätsstr. 27
93040 Regensburg
Germany
roland.meyer@sprachlit.uni-regensburg.de
http://www-slavistik.uni-r.de/institut/meyer

Refining Queries on a Treebank with XSLT Filters. Approaching the Universal Quantifier *

George Smith

Universität Potsdam

This paper discusses the use of XSLT stylesheets as a filtering mechanism for refining the results of user queries on treebanks. The discussion is within the context of the TIGER treebank, the associated search engine and query language, but the general ideas can apply to any search engine for XML-encoded treebanks. It will be shown that important classes of linguistic phenomena can be accessed by applying relatively simple XSLT templates to the output of a query, effectively simulating the universal quantifier for a subset of the query language.

1 Introduction

In the TIGER treebank (Brants et al., 2002), syntactic structure is encoded via restricted directed acyclic graphs, henceforth syntax graphs. These are tree-like structures with potentially crossing branches and labelled edges. The corpus is available in XML format. The search engine TIGERSearch (König et al., 2003) was developed to enable linguists with no previous experience in the use of either query languages or XML-encoded corpora to work with the corpus. In TIGERSearch, the encoded structures are presented to the user in a graphic representation familiar to this particular constituency. A specially designed language (König and Lezius, 2003) allows the user to query the treebank using concepts already familiar to linguists: immediate dominance, linear precedence and derived relations. The formulation of a query involves describing desired structural characteristics. The result of a query is the set of all structures in the corpus which have those characteristics.

* I would like to thank Esther König and Wolfgang Lezius for numerous helpful discussions relating to the TIGER query language. This paper is dedicated to Peter Eisenberg on the occasion of his 65th birthday.

Interdisciplinary Studies on Information Structure 02 (2005): 117–128

Dipper, S., M. Götze and M. Stede (eds.):

Heterogeneity in Focus: Creating and Using Linguistic Databases

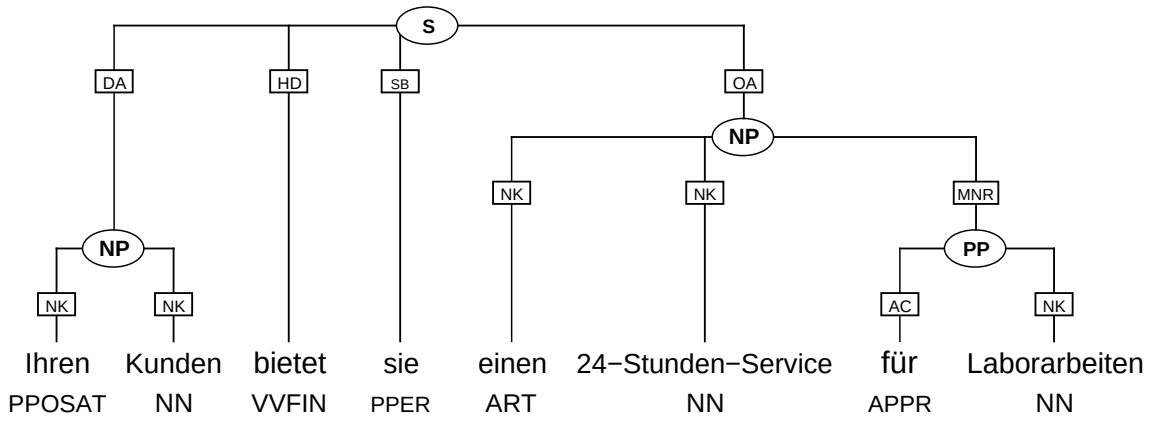


Figure 1: Three Place Verb

2 Structural Search

This section will demonstrate the importance of a universal quantifier in a tree-bank query language. A brief overview of one of the main mechanisms for searching for syntactic structure using the TIGER language will be provided in 2.1. One important area of syntax which could be more easily investigated via a universal quantifier will be given in 2.2 and 2.3.

2.1 Immediate Dominance

This section will provide a brief overview on searching syntactic structures involving immediate dominance.

(1) Ihren Kunden bietet sie einen 24-Stunden-Service für Laborarbeiten
her customers offers she a 24-hour service for lab work
'She offers her customers a 24-hour-service for lab work.'

The sentence in example (1) has the graphical representation in figure 1. A user could be interested in various structural characteristics of the sentence.

Several simple queries which match structures in the sentence are given in (2).

(2) a. `[cat="S"] >SB [pos="PPER"]`
b. `[cat="NP"] >MNR [cat="PP"]`
c. `[cat=("S"|"VP")] >DA []`

The query (2-a) matches all sentences in which the subject is a personal pronoun. This is accomplished by describing two nodes and a relation between them. The expression `[cat="S"]` describes a node with the value `S` (sentence) for the feature `cat` (category). The expression `[pos="PPER"]` describes a node with the value `"PPER"` (pronoun, personal) for the feature `pos` (part of speech). The operator `>` defines a relation of immediate dominance in which the node described to the left dominates the node described to the right. This relation is labelled `SB` (subject). The label indicates the function which the child node has within the constituent formed by the parent node.

Similarly, the query (2-b) matches all structures in which a noun phrase has a prepositional attribute, that is, in which a node with the value `"NP"` (noun phrase) for the feature `cat` immediately dominates a node with the value `"PP"` (prepositional phrase) for the feature `cat` and the relation of immediate dominance between them is labelled `MNR` (“modifier nominal right”: modifier of a noun, to the right).

The final query (2-c) matches any structure in which a node with either the value `"S"` or `"VP"` (verb phrase) for the feature `cat` dominates a node which is not further specified, and the relation of immediate dominance between them is labelled `DA` (dative), indicating that the child node functions as a dative argument within the constituent formed by the parent.

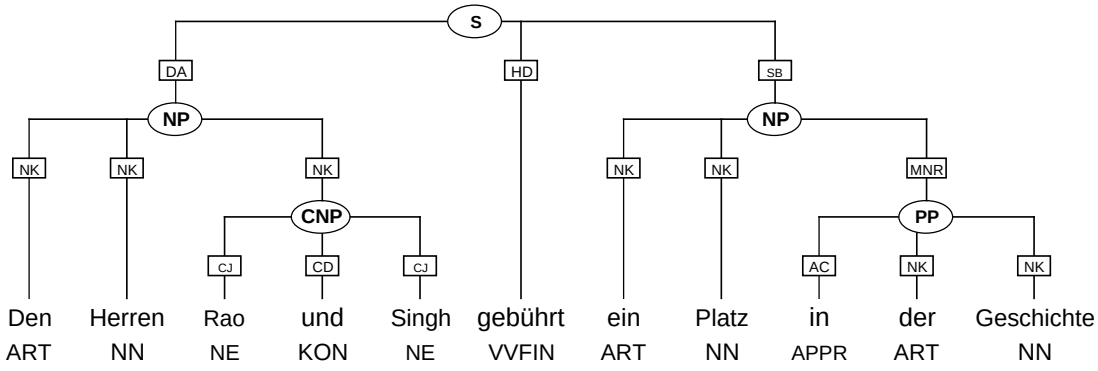


Figure 2: Two Place Dative Verb

2.2 Argument Structure and Agentivity

This section will discuss the argument structure of two classes of German verbs with regard to agentivity, preparing the argument made in section 2.3 that a universal quantifier is an important part of a treebank query language.

(3) Den Herren Rao und Singh gebührt ein Platz in der Geschichte
the Messrs Rao and Singh deserve a place in the history
'Messrs Rao and Singh deserve a place in history.'

Let us now take a look at another sentence, given in (3) with the graphical representation in figure 2, which bears certain similarities to the sentence in (1). Both sentences contain an argument in dative, which can be accessed by query (2-c). They also exhibit important differences with regard to argument structure, which will be explored here with reference to ideas presented in Primus (1999) and Eisenberg (2004), in which arguments are seen as having varying degrees of agentivity and patientivity. Agentive and patientive properties of an argument are determined compositionally using a set of thematic roles presented here in order of declining agentivity: *control*, *cause*, *move*, *exper*, *possess*.

Example (1) has the prototypical structure of a sentence with a three-place verb from the semantic field of giving and taking. The argument with the highest degree of agentivity is encoded in nominative. It is a prototypical agentive subject, excercising control, causing movement. The argument encoded in accusative is a prototypical patient. The argument encoded in dative has a low degree of agentivity, here exibiting a potential for possession. This weak agent is a prototypical recipient.

Example (3) on the other hand has a rather different argument structure. There is no argument with strong agentive properties, excercising control, or being a cause. The argument with the highest degree of agentivity is a possessor. This weak agent or recipient is again encoded in dative. The argument with no agentive properties, the patient, is encoded in nominative, avoiding a sentence with no subject. Interesting classes of German verbs have this type of argument structure, with a patientive subject and an argument in dative exhibiting a degree of agentivity compatible with the recipient role, one important group being the psychological verbs, which have the slightly more agentive *exper* encoded in dative.

2.3 The Need for the Universal Quantifier

While the query in (2-c) is sufficient for a user who simply wishes to find arguments which are dative recipients as it will access both (1) and (3), a user who wishes to capture the differences between the argument structures of the two sentences will need to further refine the query. To search for sentences with three-place verbs such as in (1) is simple.

(4)
```
#p:[cat=("S"|"VP")] >DA [] &
#p >OA []
```

The query in (4) uses a variable #p to extend the query (2-c). It specifies that there is a node #p which immediately dominates one node which has the function DA as well as another which has the function OA (object, accusative).

(5)
```
#p:[cat=("S"|"VP")] >DA [] &
#p !>OA #c
```

It is then possible to specify the presence of an argument in acccusative. In the current implementation of the TIGER language it is, however, not possible to specify the absence of one. Simply negating this relation of dominance, as in (5), results in a query stating that in addition to the dative dominated by #p, there is also a node #c, and that the specified relationship of labelled immediate dominance does not hold between #p and #c. The node #p may dominate the node #c, in which case the edge label must not be OA, or #p may not dominate #c at all. It can be any node in the tree for which the specified labelled dominance relation between #p and #c does not hold.

One option for accessing such structures might be to search using predefined classes of verbs which can have the specified argument structure. In TIGERSearch it is in principle possible to define a set of lexemes in a template (König et al., 2003) and then use that template in a search. But this method would be indirect at best, not taking the potential for multivalency into account. It also presupposes that the user already has a list of all verbs in the corpus which can have the relevant argument structure, whereas the user may well be interested in finding verbs which can have that structure. Besides, there are other important types of structures which can only be specified by stating that a constituent of a particular type has no children with particular characteristics. Examples would be sentences with no subject, including the impersonal passive, and noun phrases which are lacking a determiner, or have no attributes of a specific type, etc. There is clearly a need for a general mechanism which can accomplish this.

(6)
```
exists #p: forall #c: ((#p > #c) =>
                       (#p !>OA #c))
```

The most comfortable option for the user would be a possible extension of the TIGER language which is discussed in König et al. (2003), the addition of the universal quantifier, exemplified here in (6). This query specifies that there exists a node `#p` such that for all nodes `#c` which are immediately dominated by `#p`, the dominance relation is not labelled `OA`. The computational expense incurred by the universal quantifier is noted as the reason why it has not (yet) been included in the language. It is indeed clear that efficiently implementing a full universal quantifier as a part of the language would be difficult and time consuming. Another option, discussed in section 3, would be to filter the results of user queries such that the undesired structures are removed.

3 Simulating the Universal Quantifier with XSLT Stylesheets

This section will discuss the use of XSLT stylesheets to filter the results of user queries as a means of finding structures which would otherwise require that a universal quantifier apply to a relation of immediate dominance. XSLT was chosen for two reasons: The filtering can be accomplished with a single template and a varying test. TIGERSearch already includes an interface for piping the output of queries through XSLT. The XML representation of the syntactic structures involved will be described in 3.1. Specific filters will be described in 3.2.

3.1 The XML Representation of Syntactic Structure

In this section we will explore those elements of the XML output of a search which are relevant for further processing via an XSLT filter.

(7)
```
#p:[cat=("S"|"VP")] >DA #da
```

(8)
```
<match subgraph="s60_505">
  <variable name="#p" idref="s60_505" />
  <variable name="#da" idref="s60_503" />
</match>
```

The query in (7) is a variation of the query (2-c) from section 2.1 with variables. If we run this query on the TIGER corpus and export the results as an XML file we find, among other structures, a set of matches. These matches are encoded as in (8). A match element contains a pointer to the matching subgraph (the `subgraph` attribute). Here we see that the element `match` has children of type `variable`. These elements contain pointers to respective nodes of constituent structure (the `idref` attribute). The matching nodes are given in (9) and (10)

(9)
```
<nt id="s60_505" cat="S">
  <edge label="DA" idref="s60_503" />
  <edge label="HD" idref="s60_6" />
  <edge label="SB" idref="s60_504" />
</nt>
```

(10)
```
 <nt id="s60_503" cat="NP">
   <edge label="NK" idref="s60_1" />
   <edge label="NK" idref="s60_2" />
   <edge label="NK" idref="s60_500" />
 </nt>
```

The individual non-terminal nodes have an atttribute `cat` which encodes their grammatical category, as well as child elements of type `edge`, which represent the edges pointing to their respective child nodes. The `edge` nodes themselves

have an `idref` attribute which points to the respective child nodes, as well as a `label` attribute, which indicates the function of the child node within the constituent formed by the parent node.

3.2 Filtering Matches with XSLT

This section will describe the use of an XSLT template which functions as a filter, blocking matches which do not meet certain requirements. First we will examine filters which remove matches in which a node `#p` has children with a particular syntactic function, then we will examine a filter which removes matches which have children with a particular syntactic category.

(11)
```
<xsl:variable name="test">
  0=count(key('idkey',
              variable[@name='#p']/@idref)
             /edge[@label='OA'])
</xsl:variable>
```

(12)
```
<xsl:template match="match">
  <xsl:if test="xalan:evaluate($test)">
    <xsl:apply-templates  select="ancestor::s"
                          mode="print">
    <xsl:with-param  name="matchroot"
                     select="@subgraph"/>
    </xsl:apply-templates>
  </xsl:if>
</xsl:template>
```

The template in (12) could be applied to the output of the query (7). It filters out matches in which the node `#p` has children which function as `OA`. Undesired results are filtered by applying a test `$test` before printing. Examples of XSLT

stylesheets that print sentences are included with TIGERSearch (König et al., 2003). The work here is done in the filter[1]. The filter checks to make sure that the node pointed to by `#p` (namely `variable[@name='#p']/@idref`) has no children of type `edge` with the value `OA` for the attribute `label`. The test string can then be varied with regard to the name of the variable to be checked and the function(s) to be excluded. A variation on the test which would also exclude matches with object clauses (`OC`) is given in (13).

(13)
```
<xsl:variable name="test">
  0=count(key('idkey',
              variable[@name='#p']/@idref)
             /edge[@label='OA'or @label='OC'])
</xsl:variable>
```

Filters removing matches in which a node has no children with a particular syntactic category are more complex due to the need for an additional pointer.

(14)
```
<xsl:variable name="test">
  0=count(key('idkey',
              (key('idkey',
                   variable[@name='#p']/@idref)
                  /edge/@idref))
              [@pos='ART'])
</xsl:variable>
```

The filter in (14) could be applied to the matches of a query in which `#p` is bound to nodes with the syntactic category NP, to filter out noun phrases which do not have an article. The inner pointer is structured analog to that in (11) and (13). The outer pointer locates child nodes. The predicate `[@pos='ART']`

[1] The use of the Xalan extension function `evaluate` is not crucial here, but does make the code more modular and thus easier for the inexperienced user to modify.

locates those children with the value ART (article) for the attribute pos.

(15)
```
exists #p: forall #c: ((#p > #c) =>
                       (#c:[pos!="ART"]))
```

At this point it becomes excruciatingly clear that the restriction that a node have no children with a particular syntactic category or a particular syntactic function is far better stated with a representation as in (15), in the type of representation envisioned by König et al. (2003) than it is in raw XML.

4 Conclusion and Directions for Further Research

This paper has shown that relatively simple XSLT stylesheets are capable of providing important functionality needed by linguists interested in types of syntactic structure best described by stating that a node of a particular type has no children of a particular type or with a particular function. While the XML structures and XSLT code presented here is simple from a programming standpoint, the constituency for whom TIGERSearch was developed generally lacks the experience in computer science necessary to formulate or even modify example stylesheets. Indeed, the use of data abstraction, the graphical representation of syntactic structure as opposed to the raw XML representation, as well as the development of a specialized query language based on linguistic concepts as opposed to suggesting that linguists access the corpus via a generic XML solution such as XPath, XSLT or XQuery, was predicated on the idea that a treebank can only gain widespread use within the linguistic community if that community can query the treebank using tools it is comfortable with.

Further research could be directed toward building a graphical user interface which would take expressions formulated in the representation of the universal quantifier described in König et al. (2003) and create an XSLT template filter on the fly. This could be a stand alone application, or it could be integrated in

TIGERSearch. While extending the universal quantor to the immediate dominance relation would be by far the most useful type of extension, the idea could be expanded to other relations, such as the relation of linear precedence. This type of a solution would be less than the more elegant solution involving a full implementation of the universal quantifier, but it would provide a good deal of functionality and would be easier to implement.

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*, Sozopol, 2002.

Peter Eisenberg. *Grundriß der deutschen Grammatik. Der Satz*. Metzler, Stuttgart, 2nd edition, 2004.

Esther König and Wolfgang Lezius. The TIGER language - a description language for syntax graphs, Formal definition. Technical report, 2003.

Esther König, Wolfgang Lezius, and Holger Voormann. *TIGERSearch User's Manual*. IMS, University of Stuttgart, Stuttgart, 2003. URL `http://www.tigersearch.de`.

Beatrice Primus. *Cases and Thematic Roles. Ergative, Accusative and Active*. Niemeyer, Tübingen, 1999.

George Smith
Universität Potsdam
Institut für Germanistik
Postfach 601553
14415 Potsdam
Germany
smithg@rz.uni-potsdam.de
http://www.uni-potsdam.de/u/germanistik/ls_dgs/gs/

Exploring Lexical Patterns in Text: Lexical Cohesion Analysis with WordNet

Elke Teich[*], *Peter Fankhauser*[**]

Technische Universität Darmstadt[*], FhG-IPSI, Darmstadt[**]

We present a system for the linguistic exploration and analysis of lexical cohesion in English texts. Using an electronic thesaurus-like resource, Princeton WordNet, and the Brown Corpus of English, we have implemented a process of annotating text with lexical chains and a graphical user interface for inspection of the annotated text. We describe the system and report on some sample linguistic analyses carried out using the combined thesaurus-corpus resource.

1 Introduction

In recent years, there has been an increasing activity in building up corpora annotated at multiple linguistic levels (syllable, word, clause, text) and strata (phonology, grammar, semantics). With the growing interest in such *multi-layer corpora* comes the need for tools that support corpus annotation and exploration of the resulting annotations as well as facilitate further computational processing.

For the lower levels of the linguistic system (grammatical units, such as words, phrases, clauses), there are plenty of tools that provide the necessary functionalities. For instance, at the stratum of grammar, part-of-speech tagging and shallow phrase structure parsing can be carried out automatically at reasonable accuracy, with hardly any human intervention. Also, there are some rather mature tools for corpus inspection, such as special-purpose query (e.g., CQP (Christ, 1994a)), TIGERSearch (Lezius and König, 2000; König and Lezius, 2003), concordancers (e.g., XKwic (Christ, 1994b)) and browsers for tree structures (e.g., Annotate (Plaehn and Brants, 2000)).

Interdisciplinary Studies on Information Structure 02 (2005): 129–145
Dipper, S., M. Götze and M. Stede (eds.):
Heterogeneity in Focus: Creating and Using Linguistic Databases

However, when it comes to the unit of *text* and the analysis of *meaning*, the situation is difficult in two respects. First, fully automatic annotation is often not possible; second, tools supporting annotation and exploration exist only for selected aspects of textual analysis, e.g., for rhetorical structure (O'Donnell, 1997). Rhetorical structure is clearly an important aspect of a text's organization and vital for a full-blown interpretation of a text. But there are many other meaning-creating features in a text, which are interesting from the viewpoints of both linguistic theory and computational-linguistic processing. One such feature is *cohesion*.

1.1 Corpora Annotated for Cohesion: Motivation, Goals, Tools

Cohesion is defined as the set of linguistic means we have available for creating *texture* (Halliday and Hasan, 1976, 2), i.e., the property of a text of being an interpretable whole (rather than unconnected sentences). Cohesion occurs "where the interpretation of some element in the text is dependent on that of another. The one presupposes the other, in the sense that it cannot be effectively decoded except by recourse to it." (Halliday and Hasan, 1976, 4).

The most often cited type of cohesion is *reference*.[1] Consider example (1) (from Halliday and Hasan, 1976, 2).

(1) Wash and core six cooking *apples*. Put *them* into a fireproof dish.

In example (1), it is the cohesive tie of coreference between *them* and *apples* that gives cohesion to the two sentences, so that we interpret them as a text. The detection of such referential ties is clearly essential for the semantic interpretation of a text. Corpora annotated for reference relations are thus of interest for both linguistics, e.g., for testing theories of information structure (loci

[1] Also known as *coreference* or *anaphora* and often taken to include substitution and ellipsis, i.e., *one*-anaphora and zero-anaphora.

of high/low informational load, informational statuses (Given/New)), and computational processing, e.g., for applications such as information extraction or information retrieval.

Another type of cohesion, coacting with reference to create texture, is *lexical cohesion* (cf. Halliday and Hasan, 1976). Lexical cohesion is the central device for making texts hang together experientially, defining the aboutness of a text (cf. Halliday and Hasan, 1976, chapter 6). Typically, lexical cohesion makes the most substantive contribution to texture: According to Hasan (1984) and Hoey (1991), around fourty to fifty percent of a text's cohesive ties are lexical.

In its simplest incarnation, lexical cohesion operates with *repetition*, either simple string repetition or repetition by means of inflectional and derivational variants of the word contracting a cohesive tie. The more complex types of lexical cohesion work on the basis of the semantic relationships between words in terms of *sense relations*, such as synonymy, hyponymy, antonymy and meronymy (cf. Halliday and Hasan, 1976, 278–282). See examples of a meronymic relation (highlighted in italics) and an antonymic relation (highlighted in bold face) in (2) below; the latter at the same time is a case of repetition.[2]

(2) Tone languages use for **linguistic** contrasts *speech* parameters which also function heavily in **non-linguistic** use. [...] The problem is to disentangle the **linguistic** parameters of *pitch* from the co-occurring **non-linguistic** features.

In a text, potentially any occurrence of repetition or relatedness by sense can form a cohesive tie; but not every instance of semantic relatedness between two words in a text does necessarily create a cohesive effect. For example, if a word *linguists* occurring in sentence 1 of a text containing eighty sentences is

[2] The example is taken from text j34 of the Brown corpus.

repeated in sentence 76, a cohesive effect is rather unlikely. Also, there seem to be stronger cohesive effects involving the register-specific vocabulary rather than the “general” vobulary (cf. Section 3).

Detailed manual analyses of small samples of text (e.g., Hoey, 1991) can bring out some tendencies of how lexical cohesion is achieved; but in order to arrive at any generalizations, large amounts of texts annotated for lexical ties are needed. Manual analysis is very labor-intensive, however, and the level of inter-annotator agreement is typically not satisfactory. Thus, an automatic procedure is called for. Fortunately, lexical cohesion analysis is a suitable candidate for automization: Texts systematically make use of the semantic relations between words and detecting lexical cohesive ties simply means checking the relatedness of words in a text against a thesaurus or thesaurus-like resource. A few additional constraints must be added to arrive at plausible lexical chains, such as, e.g., the afore mentioned distance between words in a text or the specificity of the vocabulary (see also Section 2).

Automatic lexical cohesion analysis has been applied in computational linguistics for automatic text summarization (e.g., Barzilay and Elhadad, 1997). Our own motivation for building a system that automatically annotates text in terms of lexical cohesion has been to be able to explore the workings of lexical cohesion in more detail, asking questions such as (cf. Fankhauser and Teich, 2004): In a given text, what are the dominant lexical chains (indicating what the text is mainly about)? Are there differences in the strength of lexical cohesion according to the register and/or genre of a text? In a given register/genre, are there any patterns of lexical cohesion (e.g., hyponymy-hypernymy, holonymy-meronymy) that occur significantly more often than others? Can the internal make-up of lexical chains tell us anything about the genre of a text (e.g., narrative vs. factual)?

1.2 Summary; Overview of Paper

With the growing interest in richly annotated corpora, there is an increasing need for tools supporting annotation as well as exploration of corpus resources, both for linguistic and for computational purposes. The corpus processing of grammatical units is pretty well understood, but there are many unresolved issues when it comes to processing corpora at the level of text. The system we present in this paper addresses one such issue, namely the annotation and exploration of lexical cohesion.

Section 2 introduces our approach to annotation of lexical cohesion and describes the functionalities of the system. Section 3 provides some examples of linguistic analysis that we have carried out using the data generated by our system. Finally, we conclude with a summary and outlook on future research (Section 4).

2 Automatic Analysis of Lexical Cohesion

The basic means for lexical cohesion analysis are so called lexical chains, which consist of words that are related by a lexically cohesive tie. Using the SEMCOR version of the Brown Corpus, which is sense tagged with so called synsets from the Princeton WordNet (version 1.6), these ties can be determined by navigating along the relationships (synonymy, hypernymy, hyponymy, antonymy, and various kinds of meronymy) in WordNet. In addition to the direct relationships we also take into account indirect relationships, including transitive hypernymy, hyponymy, and meronymy, co-hypernymy, and co-meronymy, and ties observable directly from the text, including repetition of lemmas and of proper nouns. A more detailed description of the resources and the processing steps is given in Fankhauser and Teich (2004).

Not all the ties automatically determined in this way are necessarily cohe-

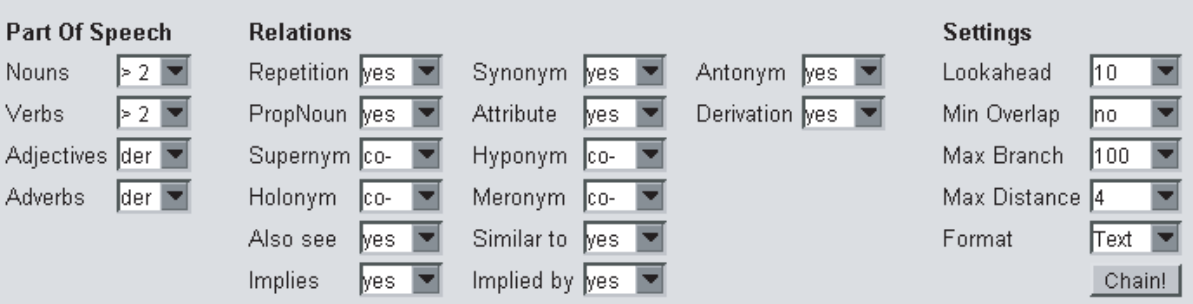


Figure 1: Options for cohesion analysis

sive. A number of factors can help in ruling out non-cohesive ties:

- Specificity and part-of-speech: A specific noun like *tone_system* is more likely to contract a lexically cohesive tie than a general verb like *be*.
- Kind of the semantic relationship: Repetition and synonymy form stronger ties than hypernymy or meronymy.
- Strength of the relationship: The direct hypernym *phonologic_system* forms a stronger cohesive tie with *tone_system* than the remote hypernym *system*.
- Distance in text: Words with many intervening words, sentences, or paragraphs are less likely to contract a cohesive tie than close words.

Our system allows fine-tuning these factors as shown in Figure 1.

The depicted settings (Part Of Speech) take only into account ties between specific nouns and verbs, which are at least at depth 3 in the WordNet hypernymy hierarchy, and include adjectives and adverbs only if they are directly related to an included noun or verb. Moreover, ties may not span more than 10 sentences (Lookahead), and transitive relationships may comprise at most 4 steps (Max Distance) with a branching factor of at most 100 alternative paths

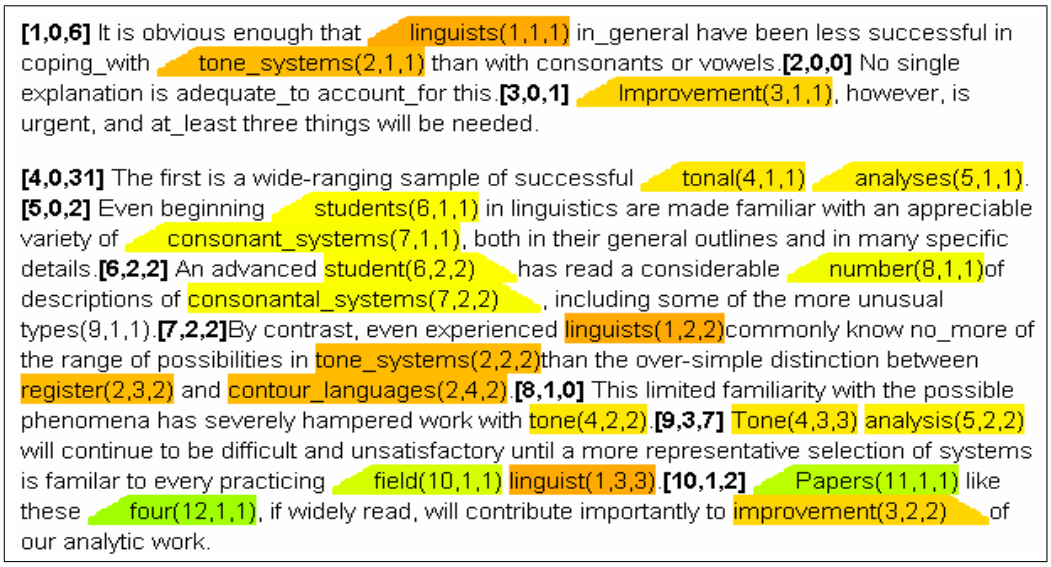


Figure 2: Text view on annotated text

(Max Branch). The kinds of relationships are not further constrained in the example setting.

Lexical chains can then be inspected from three perspectives. In the *text view* (Figure 2), each lexical chain is highlighted with an individual color, in such a way that chains starting in succession are close in color. In addition, for each sentence its number, the number of preceding sentences and the number of following sentences with a word in the same chain are given. This view can give a quick grasp on the overall topic flow in the text to the extent that it is represented by lexical cohesion.

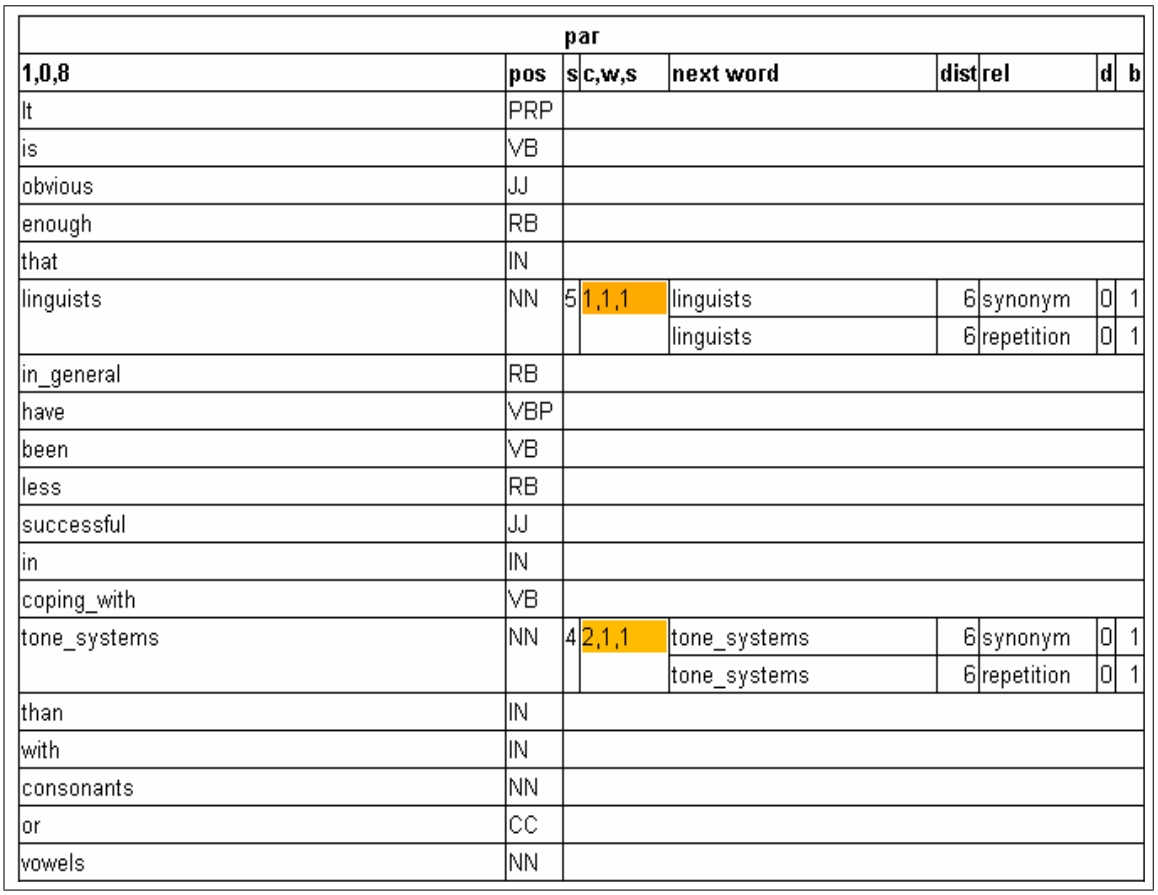
The *chain view* (Figure 3) presents chains as a table with one row for each sentence, and a column for each chain ordered by the number of words contained in it. In addition, each chain gives its most frequent word (*domwf*), and the absolute and relative number of kinds of relationships forming a tie (*repsyn* for repetition with synonymy, *rep* for repetition without synonymy, etc.). This view also reflects the topical organization fairly well by grouping the dominant chains closely.

#s, #w	22, 30	23, 28	10, 14	11, 11	5, 7	6, 7	6, 6	4, 6	5, 5	3, 5
domwf	tone	morphophonemics	phonemic_system	orthography	intonation	rule	field	theory	linguist	tone_system
repsyn	16 (55,2%)	6 (27,3%)	1 (8,3%)	5 (55,6%)	6 (100%)	6 (100%	5 (100%	5 (100%)	4 (100%)	1 (25%)
rep	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
syn	0 (0%)	1 (4,5%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
super	3 (10,3%)	0 (0%)	1 (8,3%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (25%)
hypo	2 (6,9%)	0 (0%)	1 (8,3%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
cohypo	0 (0%)	3 (13,6%)	9 (75%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (25%)
part	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (25%)
other	8 (27,6%)	12 (54,5%)	0 (0%)	4 (44,4%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
par	4	19	33	53	30	42	10	16	1	2
1									linguists	tone_systems
2										
3										
par	4	19	33	53	30	42	10	16	1	2
4	tonal									
5										
6										
7									linguists	tone_systems register contour_languages
8	tone									
9	Tone						field		linguist	

Figure 3: Chain view on annotated text

Finally, the *tie view* (Figure 4) displays for each word all its (direct) cohesive ties together with their properties (kind, distance, etc.). This view is mainly useful for checking the automatically determined ties in detail.

In addition, all views provide hyperlinks to the WordNet classification for each word in a chain to explore its semantic neighborhood. Moreover, some statistics, such as the number of sentences linking to and linked from a sentence, and the relative percentage of ties contributing to a chain are presented. These and some other statistics can then also be exported to a standard statistics package, such as MS Excel or SPSS.



par								
1,0,8	pos	s	c,w,s	next word	dist	rel	d	b
It	PRP							
is	VB							
obvious	JJ							
enough	RB							
that	IN							
linguists	NN	5	1,1,1	linguists	6	synonym	0	1
				linguists	6	repetition	0	1
in_general	RB							
have	VBP							
been	VB							
less	RB							
successful	JJ							
in	IN							
coping_with	VB							
tone_systems	NN	4	2,1,1	tone_systems	6	synonym	0	1
				tone_systems	6	repetition	0	1
than	IN							
with	IN							
consonants	NN							
or	CC							
vowels	NN							

Figure 4: Tie view on annotated text

3 Exploring lexical cohesion

On the basis of the annotated data, we have generated some statistics concerning the average **chain lengths** (in no. of sentences/words participating in a chain), according to register, of both all the chains and the dominant (i.e., the longest) chains and the distribution of **types of lexical cohesion** (repetition, synonymy, hyponymy, etc.) according to register.

As will be seen, the dominant chains in a text give a good indication of a text's topic; also, the distribution of types of lexical cohesion turns out to be a possible measure for discriminating between registers.

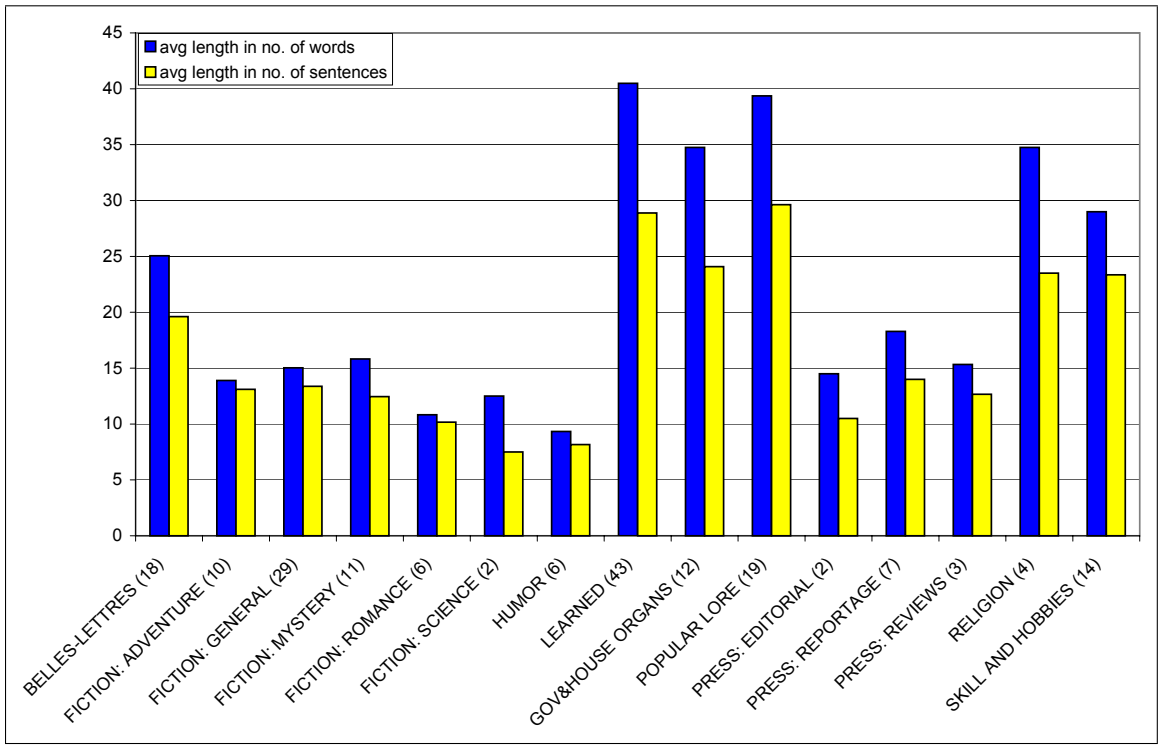


Figure 5: Average length of dominant chains by register

3.1 Chain Length

Comparing registers, the average length of lexical chains does not show substantial differences at a first glance. Most registers average between 3 and 4.5 in terms of the number of words participating in a chain and between 3 and 4 in terms of the number of sentences a chain stretches over. This means that the texts in the corpus are similarly cohesive.

However, when we compare the average length of the dominant chains across registers (i.e., the longest chains), two groups of registers stand out (cf. Figure 5): texts from the registers of LEARNED, GOVERNMENT & HOUSE ORGANS and RELIGION have relatively long dominant lexical chains and texts from PRESS and FICTION have relatively short dominant lexical chains. For example, the average length of the dominant chains in LEARNED is 40, in FIC-

TION:GENERAL it is only 15.[3]

When we look at the concrete words that make up the dominant chains, we can observe that they are good indicators of the topic of a text.[4] Short chains (with few participating words) have a different function in that they "glue" a text together locally. For example in text j34 from LEARNED (see also Figure 3), the dominant chains are built around *tone* and *phonology/morphophonemics* — this places the text in the area of linguistics, in particular phonology, and it gives us the topic of the text, which is tone. The shorter chains in this text are built around, for example, groups of words such as *explanation, theory, hypothesis, assumption* or *analysis, investigation*. One hypothesis that could be derived from such observations for this particular register is that the dominant chains are built around the register-specific vocabulary and shorter chains around the "general" vocabulary (cf. also Hoey, 1991). This hypothesis would need to be tested on more data than we have available here, however, and require a proper definition of what register-specific vocabulary means.

3.2 Types of Lexical Cohesion

Among the different types of cohesion (repetition, synonymy, hyponymy/ hypernymy, meronymy/holonymy), the most frequent means employed throughout the corpus is repetition co-occurring with synonymy with over 50% (see Figure 6, rightmost bar).

However, contrasting the different registers, there are differences in the distribution of repetition, hypernymy+(co)hyponymy and meronymy. Texts from LEARNED, RELIGION, and PRESS exhibit a higher frequency of hypernymy plus

[3] For all the data discussed here, tests for significance would have to be carried out, of course. For the time being, we conceive of the analyses reported on as purely exploratory.

[4] This observation conforms to the findings of e.g., Barzilay and Elhadad (1997), who use the dominant chains as a basis for summarization. Also, the words found in dominant chains usually have high inverse document frequency, a measure used in information retrieval.

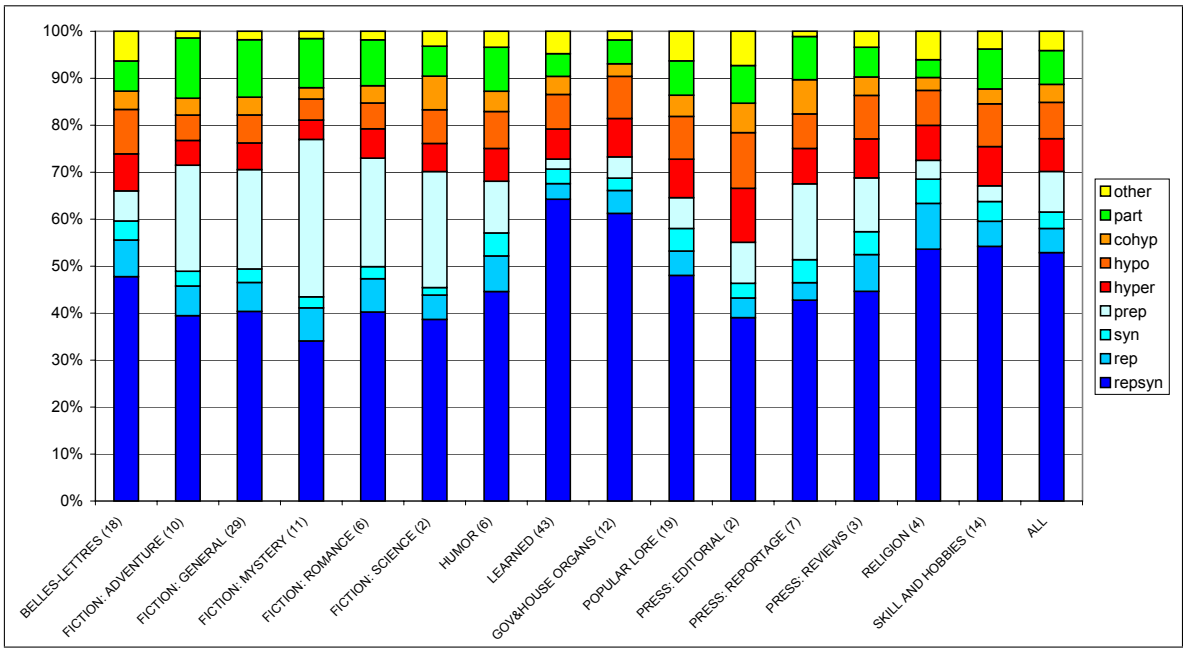


Figure 6: Types of lexical cohesion by register

(co)hyponymy than texts from FICTION. Interestingly, LEARNED and RELIGION also have the longest lexical chains relative to other registers (cf. Section 3.1). This does not come as a total surprise, however: We would expect texts from a factual genre, such as academic articles as they are included in the LEARNED register, to exhibit a strong topic continuity, whereas texts from the narrative genre, as the ones contained in the FICTION registers, can be expected to include topic shifts.

Coming back to repetition, in the LEARNED register, there is a high frequency of repetition co-occurring with synonymy, whereas in the FICTION registers repetition occurs significantly less frequently, and there is a larger amount of repetition without synonymy. This can be cautiously interpreted as follows: Texts from LEARNED try to be as unambiguous as possible, using vocabulary consistently in terms of word senses, whereas FICTION texts may actually play with ambiguity and try to be more varied in terms of vocabulary.

Finally, in the FICTION registers we encounter a substantial amount of *proper noun repetition*, which is very rare in the LEARNED register. FICTION registers also exhibit a higher frequency of meronymy. Again, this is not surprising, since fiction texts often deal with individual people who are referred to by name, and physical things, for which meronymy is more comprehensively covered in WordNet than for abstract concepts.

3.3 Summary

In summary, the findings based on the statistics presented in this section, are the following:

- **Cohesion across registers**
  - All registers included in the corpus show roughly the same degree of cohesion (where individual texts may still vary considerably in cohesive strength).
  - In different registers, cohesion is achieved by different means.
- **Cohesive patterns across registers**
  - Repetition is the most frequently used means of cohesion across registers.
  - Apart from repetition, individual registers may have a preference for a particular type of cohesion.
- **Cohesion in individual texts**
  - The dominant lexical chains (stretching over many sentences with many words participating) indicate the topic of a text.
  - In factual texts, the dominant chains tend to be made up of register-specific vocabulary.

4 Summary and conclusions

As the interest in richly annotated corpora is growing, so is the need for tools supporting annotation and exploration of multi-layer corpora. In particular, recently there is an increasing interest in the analysis of *texts*, be it for building linguistic descriptions, for testing linguistic theories or for computational applications, such as automatic summarization, text classification, information extraction or ontology building. The common interest is the interpretation of text in terms of the meaning(s) it encodes, be that rhetorical structure, information distribution or informational content.

While there is no comprehensive corpus tool available that can cater for all the linguistic needs involved in annotating text and exploring richly annotated corpus resources,[5] it has become common practice to use/build special-purpose tools that are geared to a particular annotation and/or corpus analysis task. The system we have presented in this paper is one such tool. The specific purpose it is dedicated to is to support the analysis of texts in terms of lexical cohesion. The system automatically annotates text (here: SEMCOR/Brown Corpus) in terms of lexical-cohesive ties on the basis of WordNet. The resulting annotated text can be viewed from three different perspectives, each supporting exploration of lexical-cohesive patterns from a different angle (cf. Section 2). The results of annotation can be statistically processed, simply using a standard statistics program, such as the one included in MS Excel. We have exemplified the use of some such statistics in linguistic analysis (Section 3).

With different tools taking care of different types of corpus-related tasks, special attention has to be paid to their interoperability, notably the interchange of the created corpus data. Here, the common practice now is to represent corpus resources using a standard format and data model, typically XML (see Dipper

[5] One project in this direction was the MATE project (McKelvie et al., 2001). Unfortunately, the project did not result in a scalable implementation (cf. Teich et al., 2001).

et al. (2004b) for an overview of corpus tools relying on XML). The system we have presented follows this policy, solely relying on XML and XSLT/XPath. Thus, the present research is in line with other corpus-based projects currently running or in planning, such as MULI (Baumann et al., 2004b,a), the Potsdam–Berlin SFB No. 632[6], the *Forschergruppe* at Bielefeld[7] or the project *Deutsch Diachron Digital* (Dipper et al., 2004a), only to mention a few.

In our future work, we will carry out further linguistic analyses using the data from the Brown Corpus and extend the data set to other corpora and languages (notably German). Possible applications of this research have been mentioned in passing (cf. Section 3). Notably, the data generated by our system can be used in text summarization and text classification.

Bibliography

Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL*, Madrid, Spain, 1997.

Stefan Baumann, Caren Brinckmann, Silvia Hansen-Schirra, Geert-Jan Kruijff, Ivana Kruijff-Korbayova, Stella Neumann, Erich Steiner, Elke Teich, and Hans Uszkoreit. The MULI project: Annotation and analysis of information structure in German and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisboa, Portugal, 2004a.

Stefan Baumann, Caren Brinckmann, Silvia Hansen-Schirra, Geert-Jan Kruijff, Ivana Kruijff-Korbayova, Stella Neumann, and Elke Teich. Multidimensional annotation of linguistic corpora for investigating information structure. In *Proceedings of HLT/NAACL Workshop 'Frontiers in corpus annotation'*. Human Language Technology (HLT) Conference/Annual Meeting of the North-American Chapter of the Association for Computational Linguistics (NAACL), 2004b.

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

[7] http://coli.lili.uni-bielefeld.de/Texttechnologie/Forschergruppe/

Oliver Christ. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of the 3rd Conference on Computational Lexicography and Text research (COMPLEX 94)*, pages 23–32, Budapest, Hungary, 1994a.

Oliver Christ. The IMS Corpus Workbench User Manual. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, 1994b. (`http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench`).

Stefanie Dipper, Lukas Faulstich, Ulf Leser, and Anke Lüdeling. Challenges in modelling a richly annotated diachronic corpus of German. In *Proceedings of the LREC Workshop on XML-based richly annotated corpora*, Lisboa, Portugal, 2004a.

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*, Lisboa, Portugal, 2004b.

Peter Fankhauser and Elke Teich. Multiple perspectives on text using multiple resources: experiences with XML processing. In *Proceedings of the LREC Workshop on XML-based richly annotated corpora*, Lisboa, Portugal, 2004.

MAK Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, London, 1976.

Ruqaiya Hasan. Coherence and cohesive harmony. In J. Flood, editor, *Understanding Reading Comprehension*, pages 181–219. International Reading Association, Delaware, 1984.

Michael Hoey. *Patterns of lexis in text*. Oxford University Press, Oxford, 1991.

Esther König and Wolfgang Lezius. The TIGER language — a description language for syntax graphs, formal definition. Technical report, IMS, Universität Stuttgart, Germany, 2003. (`http://www.tigersearch.de`).

Wolfgang Lezius and Esther König. Towards a search engine for syntactically annotated corpora. In Ernst G. Schukat-Talamazzini and Werner Zühlke, editors, *KONVENS-2000 Sprachkommunikation*, pages 113–116. VDE Verlag, Ilmenau, Germany, 2000.

David McKelvie, Amy Isard, Andreas Mengel, Morten Baun Møller, Michael Grosse, and Marion Klein. The MATE workbench — An annotation tool for XML coded speech corpora. *Speech Communication*, 33(1–2):97–112, 2001.

Michael O'Donnell. RST-Tool: An RST analysis tool. In *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany, 1997.

Oliver Plaehn and Thorsten Brants. Interactive corpus annotation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, 2000. Athens.

Elke Teich, Silvia Hansen, and Peter Fankhauser. Representing and querying multilayer annotated corpora. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 228–237, University of Pennsylvania, Philadelphia, 2001.

Elke Teich
Technische Universität Darmstadt
Institut für Sprach- und Literaturwissenschaft
Hochschulstr. 1 (S 1 03 / 190)
D-64289 Darmstadt
teich@linglit.tu-darmstadt.de
http://www.ifs.tu-darmstadt.de/linglit_teich/

Peter Fankhauser
Fraunhofer IPSI
Divison I-Info
Dolivostr. 15
D-64293 Darmstadt
fankhaus@ipsi.fraunhofer.de
http://www.ipsi.fraunhofer.de/~fankhaus/