

An Infrastructure for More Reliable Corpus Analysis

Kerstin Eckart, Kurt Eberle, Ulrich Heid

SFB-732, B3, Universität Stuttgart
Institut für maschinelle Sprachverarbeitung
– Computerlinguistik –
Azenbergstr. 12
D 70174 Stuttgart

WSPP2010 Workshop, Valletta, Malta

Overview

- Objectives and framework
- Support for three different pipelines of corpus-based syntactic and semantic analysis
- A database as an infrastructure
- The three pipelines: details and examples focus on reliability
- Conclusions and future work

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora
- Managing analysis results produced with different analysis processes

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora
- Managing analysis results produced with different analysis processes
- Comparing analyses
 - Computing reliability measures
 - Optimizing analysis tools

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora
- Managing analysis results produced with different analysis processes
- Comparing analyses
 - Computing reliability measures
 - Optimizing analysis tools
- Integrating analysis results

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora
- Managing analysis results produced with different analysis processes
- Comparing analyses
 - Computing reliability measures
 - Optimizing analysis tools
- Integrating analysis results
- Extracting information and including it into the knowledge base of existing analysis tools in a bootstrapping approach

Objectives

- Testing linguistic hypotheses (syntax, semantics)
⇒ Need for reliable sentence analyses on huge corpora
- Managing analysis results produced with different analysis processes
- Comparing analyses
 - Computing reliability measures
 - Optimizing analysis tools
- Integrating analysis results
- Extracting information and including it into the knowledge base of existing analysis tools in a bootstrapping approach
- Interoperability
 - with existing infrastructure and formats (ANNIS/PAULA)
 - with upcoming ISO-standards (LAF/GrAF framework)

Framework

Disambiguating German *-ung*-nominalizations

- DE *-ung*-nominals can be sortally ambiguous:
events (e) – states (s) – objects (o)
 - *Teilung* (division): e | s
 - *Abdeckung*: e (covering) | s (... being covered) | o (cover)

Framework

Disambiguating German *-ung*-nominalizations

- DE *-ung*-nominals can be sortally ambiguous:
events (e) – states (s) – objects (o)
 - *Teilung* (division): e | s
 - *Abdeckung*: e (covering) | s (... being covered) | o (cover)
- Context partners as indicators of sortal readings:
 - modifiers: *rostige Abdeckung* (rusty) \Rightarrow o
 - selectors: *unterhalb der Abdeckung* (underneath) \Rightarrow o
Abdeckung durchführen (carry out ...) \Rightarrow e

Framework

Disambiguating German *-ung*-nominalizations

- DE *-ung*-nominals can be sortally ambiguous:
events (e) – states (s) – objects (o)
 - *Teilung* (division): e | s
 - *Abdeckung*: e (covering) | s (... being covered) | o (cover)
- Context partners as indicators of sortal readings:
 - modifiers: *rostige Abdeckung* (rusty) \Rightarrow o
 - selectors: *unterhalb der Abdeckung* (underneath) \Rightarrow o
Abdeckung durchführen (carry out ...) \Rightarrow e
- Computational linguistic tasks:
 - (1) Identification of nominal-readings in context
 - (2) Acquisition of indicator data from context
 - (3) Enhancement of analyses: reliability

Pipelines

- (1) Primary Task:
Extracting readings of potentially ambiguous items from text

Pipelines

- (1) Primary Task:
Extracting readings of potentially ambiguous items from text
- (2) Knowledge Acquisition:
Enhancing knowledge bases of existing tools

Pipelines

- (1) Primary Task:
Extracting readings of potentially ambiguous items from text
 - (2) Knowledge Acquisition:
Enhancing knowledge bases of existing tools
- ⇒ (1)+(2) Bootstrapping approach for task-specific disambiguation

Pipelines

- (1) Primary Task:
Extracting readings of potentially ambiguous items from text
 - (2) Knowledge Acquisition:
Enhancing knowledge bases of existing tools
- ⇒ (1)+(2) Bootstrapping approach for task-specific disambiguation
- (3) Reliability:
Comparison and merging of analysis results
Based on the assumption: if two (or more) tools produce the same analysis result, it is more reliable than diverging analyses.

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses
- Analysis results produced by the tools

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses
- Analysis results produced by the tools
- Findings of the inspection of analysis results,
possibly produced semi-automatically

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses
- Analysis results produced by the tools
- Findings of the inspection of analysis results,
possibly produced semi-automatically
- Graph-based representations of individual analyses or inspections

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses
- Analysis results produced by the tools
- Findings of the inspection of analysis results,
possibly produced semi-automatically
- Graph-based representations of individual analyses or inspections
- Tools (or: tool versions) which produce
analyses of the primary data and which may be further developed

A database infrastructure

Requirements (1/2)

Managing all data types for the analysis of linguistic phenomena

- Primary data:
(partial) corpora, texts or single sentences
used for analyses
- Analysis results produced by the tools
- Findings of the inspection of analysis results,
possibly produced semi-automatically
- Graph-based representations of individual analyses or inspections
- Tools (or: tool versions) which produce
analyses of the primary data and which may be further developed
- Metadata:
 - Contents-related metadata, (author, language, etc.)
 - Technical metadata (character encoding, data size, etc.)
 - Metadata specifying the analysis process: tool version in use,
parameters, knowledge sources

A database infrastructure

Requirements (2/2)

Infrastructural function in linguistic research:

Constant evolution of tools, new types of data: temporal aspect

- Extensibility:
Introducing new types of data objects
without changes to the DB schema

A database infrastructure

Requirements (2/2)

Infrastructural function in linguistic research:

Constant evolution of tools, new types of data: temporal aspect

- Extensibility:
Introducing new types of data objects
without changes to the DB schema
- Theory-independence:
Data structures without a preference for a particular linguistic theory
→ Possibility to compare different analysis types

A database infrastructure

Requirements (2/2)

Infrastructural function in linguistic research:

Constant evolution of tools, new types of data: temporal aspect

- Extensibility:
Introducing new types of data objects
without changes to the DB schema
- Theory-independence:
Data structures without a preference for a particular linguistic theory
→ Possibility to compare different analysis types
- Reproducibility:
Analyses are reproducible with the same tool and data versions
→ Intermediate states in tool development can be represented

A database infrastructure

Design

- Database conceptually divided into macroscopic and microscopic layer
 - Primary data and tools are objects of macro-layer only
 - Analyses and inspections are represented on both layers
 - * as atomic objects at the macro-layer
 - * by one or more graph-based representations as structured objects of the micro-layer

A database infrastructure

Design

- Database conceptually divided into macroscopic and microscopic layer
 - Primary data and tools are objects of macro-layer only
 - Analyses and inspections are represented on both layers
 - * as atomic objects at the macro-layer
 - * by one or more graph-based representations as structured objects of the micro-layer
- Both layers mainly consist of objects and directed binary relations between them

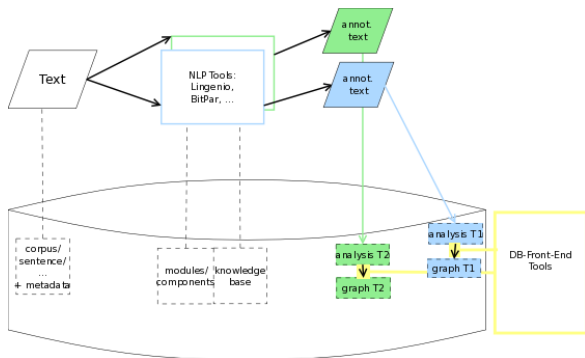
A database infrastructure

Design

- Database conceptually divided into macroscopic and microscopic layer
 - Primary data and tools are objects of macro-layer only
 - Analyses and inspections are represented on both layers
 - * as atomic objects at the macro-layer
 - * by one or more graph-based representations as structured objects of the micro-layer
- Both layers mainly consist of objects and directed binary relations between them
- Objects and relations are typed by atomic types or Boolean combinations of these
→ DB easily extensible for new types of data.

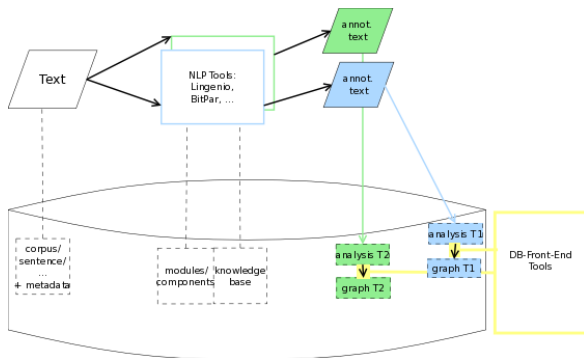
Supporting three Pipelines

(1) Extracting readings of potentially ambiguous items from text



Supporting three Pipelines

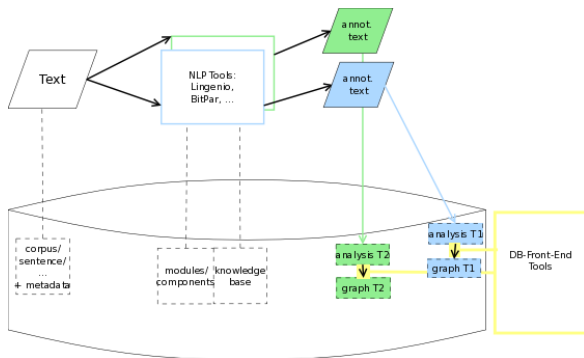
(1) Extracting readings of potentially ambiguous items from text



- Tokenizing – tagging – parsing

Supporting three Pipelines

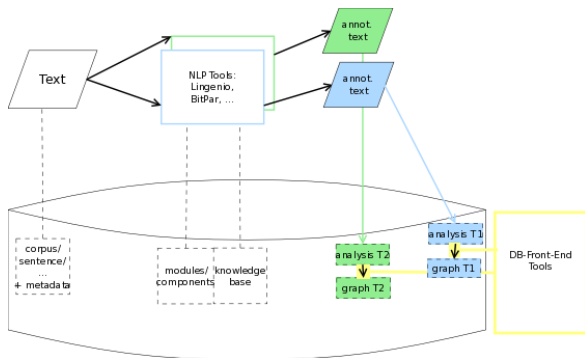
(1) Extracting readings of potentially ambiguous items from text



- Tokenizing – tagging – parsing
- Interpretation step for task-specific disambiguation: Modifiers and selectors with sortal selection restrictions

Supporting three Pipelines

(1) Extracting readings of potentially ambiguous items from text

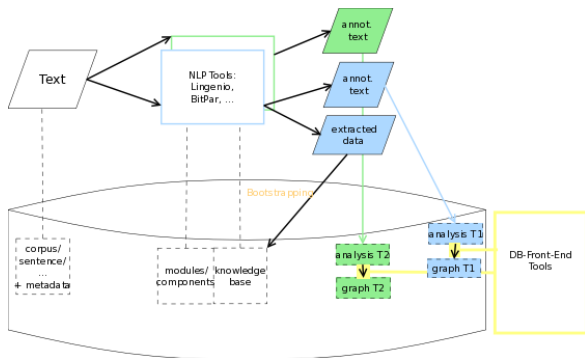


- Tokenizing – tagging – parsing
- Interpretation step for task-specific disambiguation: Modifiers and selectors with sortal selection restrictions

- Modifiers: (*Abdeckung:*) *hastig* (hasty), *schlampig* (unorderly), etc. \Rightarrow e
- Selectors:
(*Abdeckung*) *durchführen* (carry out), *beginnen*, *beenden* (start, stop) \Rightarrow e
(*Teilung*) *überwinden* (overcome) \Rightarrow s

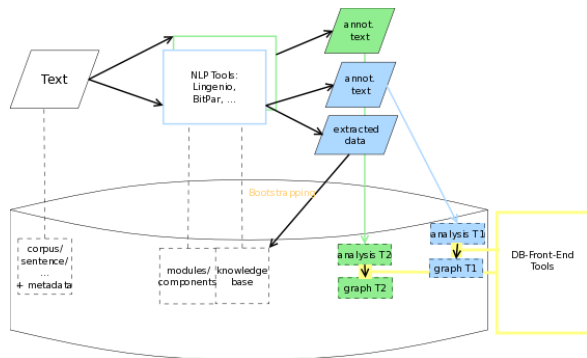
Supporting three Pipelines

(2) Enhancing knowledge bases of existing tools



Supporting three Pipelines

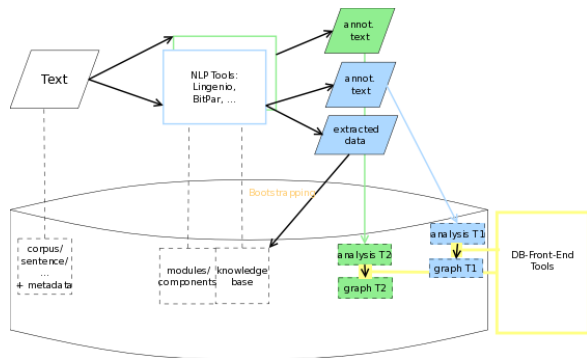
(2) Enhancing knowledge bases of existing tools



- Extraction of example sentences for a particular syntactic construction, maybe even an underspecified one
- Automatic pre-classification and manual classification of indicator candidates

Supporting three Pipelines

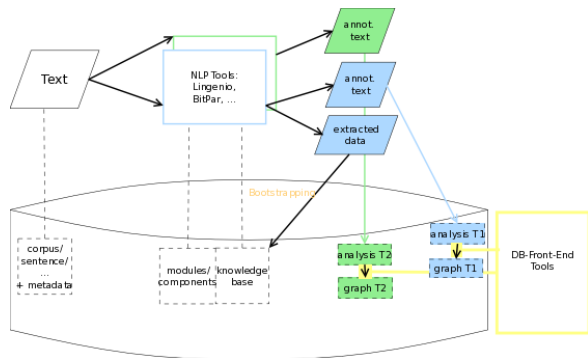
(2) Enhancing knowledge bases of existing tools



- Extraction of example sentences for a particular syntactic construction, maybe even an underspecified one
- Automatic pre-classification and manual classification of indicator candidates
- Insertion into the lexicon

Supporting three Pipelines

(2) Enhancing knowledge bases of existing tools



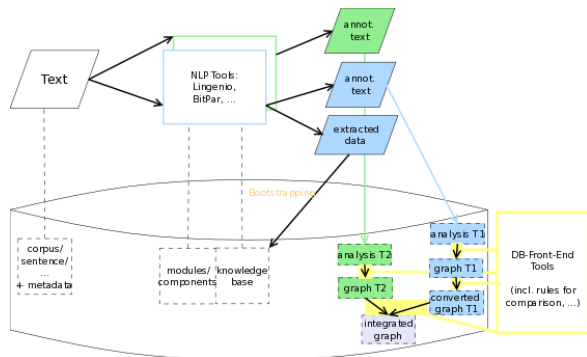
- Extraction of example sentences for a particular syntactic construction, maybe even an underspecified one
- Automatic pre-classification and manual classification of indicator candidates
- Insertion into the lexicon

$V < SUBJ_{-ung} OBJ_{dass+sentence} >$

- ... die Verschärfung der ...Gesetze ... bewirkt, dass ...
the fact that the laws have been made more strict causes that ...

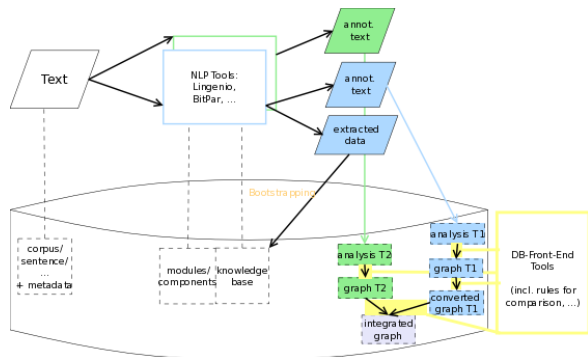
Supporting three Pipelines

(3) Comparison and merging of analysis results



Supporting three Pipelines

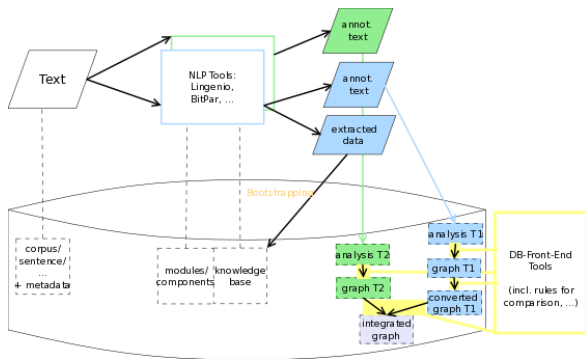
(3) Comparison and merging of analysis results



- Converting graphs into structures that can be compared

Supporting three Pipelines

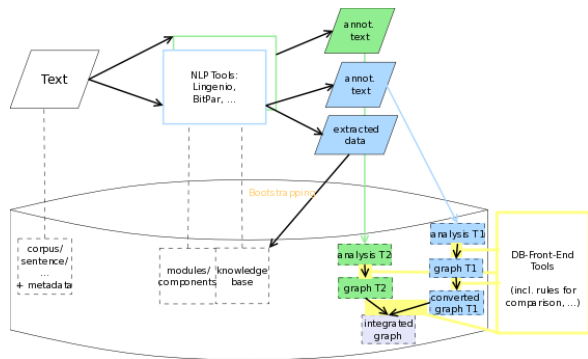
(3) Comparison and merging of analysis results



- Converting graphs into structures that can be compared
- Identification and classification of the differences and similarities

Supporting three Pipelines

(3) Comparison and merging of analysis results



- Converting graphs into structures that can be compared
- Identification and classification of the differences and similarities
- Application of merging rules

Reliability: Comparison and merging of analysis results

Example

Auch bei den CO-Werten liegen die Messungen weit unter dem zulässigen Grenzwert von 250ppm (parts [per million, Bestandteile in einer Million Teile]) . . .

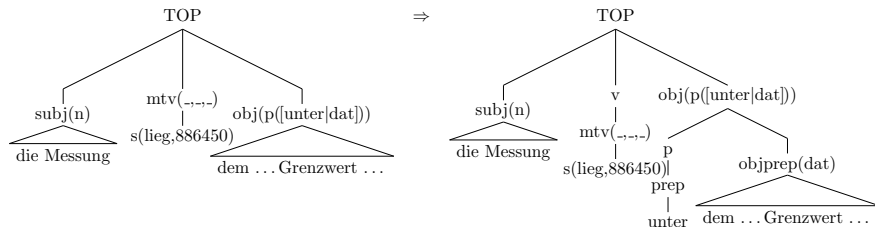
Also when it comes to the values for carbon monoxide, the measured data are much below the allowed threshold of 250 ppm [...]

- Dependency analysis by the B3-Analysis-Tool (Eberle et al. 2008)
- Constituency analysis by BitPar (Schmid 2004)

Reliability: Comparison and merging of analysis results

Converting the structure

- Conversion of B3-Analysis-Tool output into a BitPar-like format
- Rules inserting nodes and projections into the dependency structure (Eberle 2002):

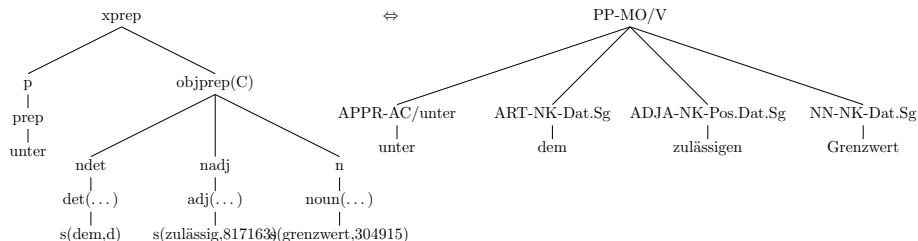


Reliability: Comparison and merging of analysis results

Merging Rules (1/3)

- Representation of the structure of PPs:
 - flat in BitPar (e.g. p det adj n)
 - structured in B3 analysis (e.g. p objprep (det adj n))

Relatable by means of a **graph equivalence rule**:

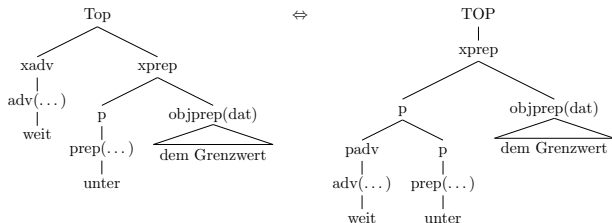


Reliability: Comparison and merging of analysis results

Merging Rules (2/3)

- Difference in attachment: *weit* ('much')
 - adverbial in the B3 analysis
 - preposition modifier in BitPar
- Minor difference concerning an embedded partial structure
- Embedding structure identical

Mapping rule for local interpretation differences:

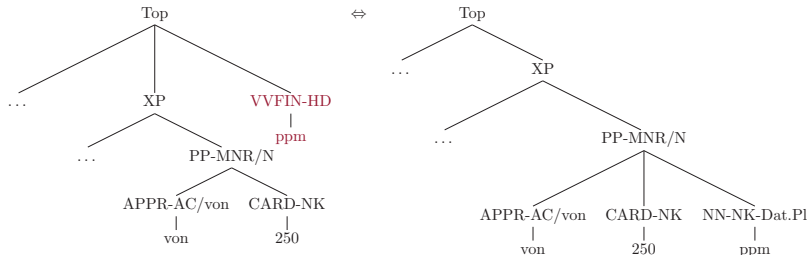


Reliability: Comparison and merging of analysis results

Merging Rules (3/3)

- Item *ppm*
 - contained in the dictionary of the B3-Analysis-Tool
 - unknown to BitPar
- If a tool has lexical information about a given item
→ tool assumed to be more reliable on that item

Rule for (resource-based) correction:



Rules for structure comparison

Summary

- Types:
 - graph equivalence (PP: deep \leftrightarrow flat)
 - mapping in case of local interpretation differences (modification of PP)
 - resource-based correction (lexicon data)

Rules for structure comparison

Summary

- Types:
 - graph equivalence (PP: deep \leftrightarrow flat)
 - mapping in case of local interpretation differences (modification of PP)
 - resource-based correction (lexicon data)
- Rules used for frequent phenomena

Rules for structure comparison

Summary

- Types:
 - graph equivalence (PP: deep \leftrightarrow flat)
 - mapping in case of local interpretation differences (modification of PP)
 - resource-based correction (lexicon data)
- Rules used for frequent phenomena
- Current state:
Rules under study,
Implementation planned for 2nd half 2010

Conclusion

- Database tool to support corpus-based linguistic research workflows:
follows typical corpus-linguistic bootstrapping spiral:
hypothesis₁ → test on data → analysis of results → hypothesis₂

Conclusion

- Database tool to support corpus-based linguistic research workflows:
follows typical corpus-linguistic bootstrapping spiral:
hypothesis₁ → test on data → analysis of results → hypothesis₂
- Database covers several dimensions:
tool(version)s – data – analyses – inspection results

Conclusion

- Database tool to support corpus-based linguistic research workflows:
follows typical corpus-linguistic bootstrapping spiral:
hypothesis₁ → test on data → analysis of results → hypothesis₂
- Database covers several dimensions:
tool(version)s – data – analyses – inspection results
- Database supports work towards more reliable analyses:
Possibility to compare analyses and to merge them
(by means of rules for structure comparison)

Future Work

- Implementation of rules of the above types, broadening of mappable fragment

Future Work

- Implementation of rules of the above types, broadening of mappable fragment
- Experiments with additional parsers:
 - FSPar (dependency parsing, Schiehlen 2003)
 - LFG

Future Work

- Implementation of rules of the above types, broadening of mappable fragment
- Experiments with additional parsers:
 - FSPar (dependency parsing, Schiehlen 2003)
 - LFG
- Weighting of analyses and voting

Information

- ANNIS/PAULA: developed at the Collaborative Research Centre 632, <http://www.sfb632.uni-potsdam.de/~d1/annis/>
- Stefanie Dipper. XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In Proceedings of Berliner XML Tage 2005, 39–50. Berlin, Germany. 2005.
- Kurt Eberle. Tense and Aspect Information in a FUDR-based German French Machine Translation System. In Hans Kamp and Uwe Reyle, editors, How we say WHEN it happens. Contributions to the theory of temporal reference in natural language, 97–148. Niemeyer, Tübingen. Ling. Arbeiten, Band 455. 2002.
- Kurt Eberle, Ulrich Heid, Manuel Kountz and Kerstin Eckart. A Tool for Corpus Analysis using partial Disambiguation and Bootstrapping of the Lexicon. Storrer, Angelika, Alexander Geyken, Alexander Siebert and Kay-Michael Würtzner (eds.): Text Resources and Lexical Knowledge (Berlin: Walter de Gruyter) 2008, 145–157. 2008.
- LAF/GrAF: ISO/DIS 24612 Language resource management - Linguistic annotation framework (LAF). 2009.
- Michael Schiehlen. A Cascaded Finite-State Parser for German. In: EACL'03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 163–166. Budapest, Hungary. 2003.
- Helmut Schmid. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In Proceedings of the 20th International Conference on Computational Linguistics, Coling'04, volume 1, 162–168, Geneva, Switzerland. 2004