Exploiting large corpora: A circular process of partial syntactic analysis, corpus query and extraction of lexicographic information

Hannah Kermes & Stefan Evert
Institut für Maschinelle Sprachverarbeitung, University of Stuttgart, Germany.

1 Introduction

Our approach follows the work of Eckle-Kohler (1999) who used a regular grammar to extract lexicographic information from text corpora. We employ a system that allows to improve her query-based grammar especially with respect to recall and speed without reducing accuracy. In contrast to Eckle-Kohler (1999), we do not attempt to parse a whole sentence or phrase at once during the extraction process, but build complex structures incrementally. The intermediate results are annotated in the corpus and used as input for following iterations.[1] This concept enables us to accommodate new aspects such as agreement information and the use of annotated structures together with their features (for partial parsing as well as for extraction purposes).

Our goal is to design a tool that can be used with both small and large corpora. In addition to partial syntactic analysis we provide queries (based on the parsing results) for an interactive use. The idea is to build up flat annotations of (maximal) syntactic constituents (noun phrases (NP), prepositional phrases (PP), adjectival phrases (AP), adverbial phrases (AdvP) and verbal complexes (VC)) incrementally, using a multi-pass algorithm. The chunks/phrases allow embedding of chunks/phrases of other categories as well as recursive embedding, but no PP attachment.

The incremental structure-building procedure enables us to analyse chunks/phrases independently of their immediate context, i.e., even if we cannot parse the whole sentence or phrase we might still parse part of it. Analyses of complex chunks/phrases have to be executed only once, the results being annotated. Consequently, even when dealing with very large corpora, interactive queries and the extraction process are relatively fast. Besides, as features specifying the character of chunks/phrases are annotated, structures having certain characteristic can be extracted easily and quickly.

We are also able to include agreement morphology in the process. Thus, we can check the content of chunks/phrases with respect to agreement features such as case, number, and gender. Agreement information of chunks/phrases is disambiguated to the extent possible without guessing and added to the structural mark-up.

2 Technical framework

Our tools are based on the IMS Corpus Workbench[2] (CWB). The CWB is an environment for storage and querying of large corpora with shallow annotations. Currently, the maximum size of a single corpus can be approximately 300 million words, depending on number and type of annotation. The CWB provides fast access to corpora by using a separate lexicon and a full index for each annotation level. The data are stored in a compact proprietary format, and compressed with specialised algorithms (Huffmann coding for the token sequence and variable-length encodings for the indices).

The CWB was initially developed for corpora annotated at token level only (typically with part-of-speech (PoS) and lemma values). Later, support for flat, non-overlapping structural annotation was added (referred to as *structural attributes*). Since this mark-up was intended for the annotation of document structure (e.g., source files, paragraphs, and sentences), the regions of structural attributes are neither hierarchical nor do they allow recursion. Besides, compression algorithms are not necessary to store the relatively small number of sentences, paragraphs, etc. in a corpus.

Queries can be specified in terms of regular expressions over tokens and their linguistic annotations, using the Corpus Query Processor component (CQP). In contrast to most CFG-based parsers, the CQP query language allows complex expressions at the basic token level. These include regular expression matching of tokens and annotated strings (optionally ignoring case and/or diacritics), tests for

[1] A similar approach was conducted by Steve Abney for English using a cascaded finite-state parser. (cf. Abney 1991 and 1999)
[2] See Christ (1994) for an overview. More information on the IMS Corpus Workbench is available from http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench .

membership in user-specified word lists, and arbitrary Boolean expressions over feature-value pairs. Additionally, "global constraints" can be used to specify dependencies between arbitrary tokens in a CQP query.

CQP includes a simple macro language, based on string replacement with interpolation of up to 10 arguments. The "body" of a macro (which is substituted for each macro "call" in a query) may contain further (non-recursive) macro invocations. Thus, complex queries can be broken down into small parts, similar to the rules of a context-free grammar. Macro definitions are loaded from text files and can be modified at run-time.

The CWB includes support for set-valued annotations encoded as string values using a special disjunctive notation. This allows treatment of agreement features, which is one of our improvements over the work of Eckle-Kohler (1999). For example, all possible combinations of case, gender, and number values a certain noun may have are stored as a single, set-valued annotation (a *feature vector*). Unification of feature values (which is the basis of most complex grammar formalisms) is equivalent to the (set-theoretic) intersection of the corresponding feature vectors. Special operators based on regular expressions are available to test for the presence of features (e.g. a noun phrase which *might* have genitive case) as well as the uniqueness of feature values (a noun phrase uniquely identified as genitive).

We considered a number of alternative approaches for the parsing stage as well, in particular those for which standard tools are available.

- Complex grammars (e.g., in the LFG or HPSG framework) can model the hierarchical structure of language, and are well-suited for handling attachment ambiguities. Drawbacks include slow parsing speed, lack of robustness, dependence on an extensive lexicon as a prerequisite, and the complex interactions between rules that complicate both grammar development and the adjustment to a particular domain seriously. Furthermore, complex grammars usually return a large number of possible analyses for each sentence, which cannot be stored and queried efficiently for large corpora. Thus, an additional, and probably rather unreliable disambiguation component or labour-intensive manual disambiguation would be necessary.

- Context-free grammars (CFGs) are modular (i.e. there is little interaction between different rules) and allow for fast parsing. In most CFG-based systems, however, modelling agreement and special (lexical) constructions requires large numbers of additional rules, which makes grammars unwieldy and slows down the parsing process. For the automatic analysis of large amounts of text, further "robustness" rules are needed. Partial or full disambiguation of agreement information is difficult to achieve.

- Probabilistic context-free grammars (PCFGs) extend the CFG formalism with a statistical model of lexical information such as subcategorisation frames and collocations. In contrast to complex grammars, probabilistic "lexicon entries" are learned from the input text and training data without human intervention. PCFG-based parsers are slower than their CFG counterparts and require a considerable amount of working memory for their large parameter sets. A particular problem for PCFGs are marked constructions and other special cases, where the parser almost inevitably prefers a more frequent unmarked alternative. In general, PCFG parsers perform a full disambiguation of agreement features involving guesswork rather than the partial disambiguation that we prefer.

To sum up, the advantages which led us to use the CWB as a framework for our tools are: (i) The possibility to work with large corpora. After compression, the surface forms and lexical annotations (lemma, etc.) require approximately 30 bits/token of disk space, whereas categorical annotations (PoS, agreement features, etc.) require 10 bits/token or less. (ii) CQP efficiently evaluates complex queries on large corpora. Disk files are accessed directly and do not have to be loaded into memory first. It is this feature which makes a multi-pass algorithm (in which CQP is frequently restarted in order to re-use intermediate results; see section 334 for details) feasible at all. (iii) The query language is modular and allows easy treatment of special cases (using additional rules, word lists, or structural mark-up of multiword entities). (iv) The same representation formalism and query language can be used at the parsing stage, for interactive querying of the final results, and for the extraction of lexical information.

3 General concept

The general concept of our approach is to combine two usually separate processes (see also Figure 1): (i) annotation of syntactic structures and (ii) extraction of linguistic information. Central to our approach is a set of hierarchical query macros. These macros serve basically two purposes: (i) they can be used by an annotator tool to analyse syntactic structure and (ii) they can be interactively used to extract linguistic information. In the first case, the results of the queries are annotated as structural mark-up in the corpus, incrementally building up larger and larger structures. In the latter case, the queries are used by a human user to extract linguistic information from the corpus. For that purpose, the queries use the structural mark-up the annotator tool constructed by means of the queries themselves.

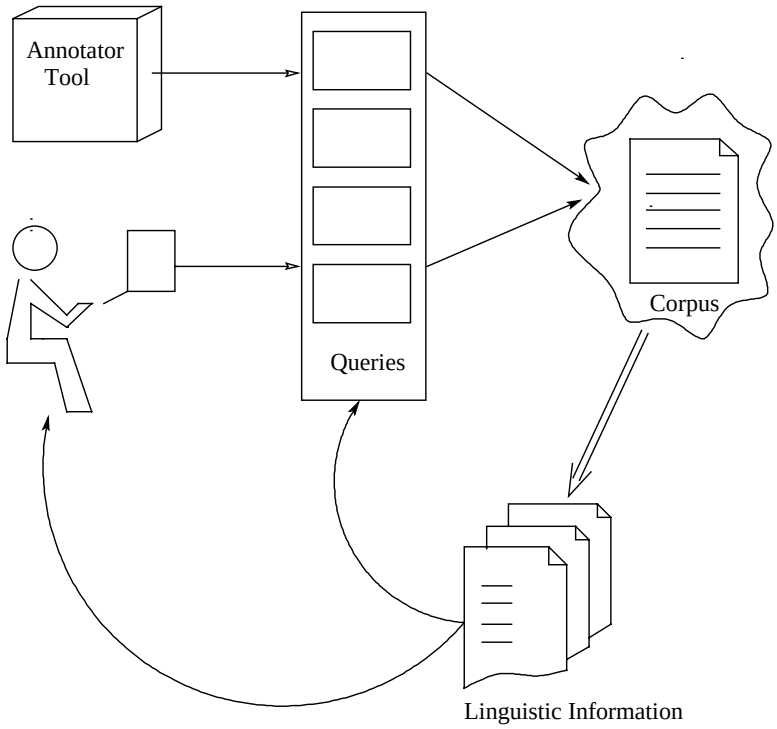

Figure 1: general concept

The results of the extraction process (in the form of linguistic or lexicographic information) may then be used either directly by the user or they may be fed back into the queries to improve them. Useful information for the latter are wordlists, subcategorization information, etc.. The annotator tool as well as the human user can then use the refined queries to optimise both the annotation and the extraction process.

4 The parser

The parser is a combination of Perl scripts and CQP queries applied in multiple passes. The results produced by CQP are post-processed by Perl scripts. These scripts check the results with respect to agreement and possibly other criteria rejecting inappropriate structures. The remaining structures are then annotated in the corpus. This procedure is repeated several times in order to build more and more complex structures. The annotated results of previous steps are taken as additional input for further steps. In general, there are two different types of passes: (i) a primary “preparing” pass that is executed only once, and (ii) a secondary pass that is run several times (see Figure 2).

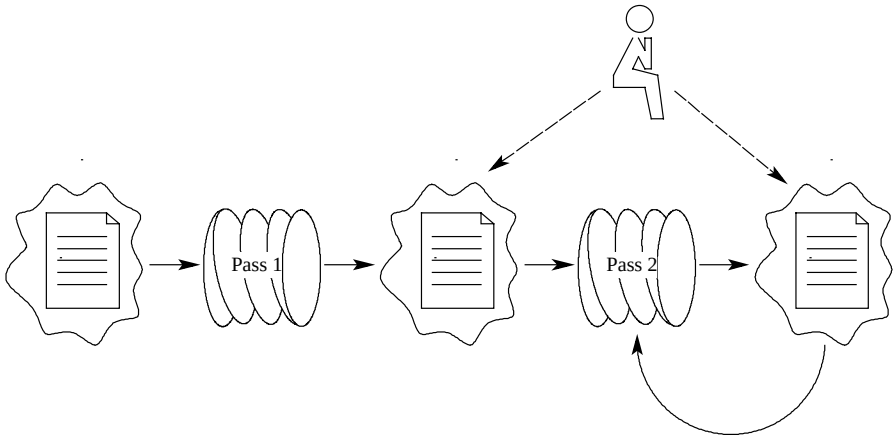

Figure 2: the parsing process

In the primary pass simple non-recursive chunks/phrases are built. These chunks/phrases are mostly syntactically and/or lexically special chunks, i.e. chunks/phrases that are either special with respect to their distribution, their grammatical function, or their structure. Collecting and annotating these chunks in a preliminary step enables us to model these chunks/phrases independently of other chunk/phrase rules, which considerably improves performance and speeds up the parsing process. The queries can be adjusted to the special characteristics (lexical or structural) of the chunks/phrases. These features are made accessible in the form of chunk annotations. Thus, in further steps these chunks/phrases may be either treated as regular chunks or according to their special character.

The secondary pass constitutes the main part of the parsing process. It is designed to run several times. The number of runs is not predetermined. Currently, we run it three times, which seems to provide a sufficient depth of embedding for most of the text in our corpora. Each iteration of the secondary pass takes the annotated structural mark-up of all previous steps (primary or secondary) as input to build more complex chunks/phrases.

The macro queries resemble the rules of a context free grammar (CFG), i.e., they state that chunks/phrases may contain a specifier, certain embedded chunks/phrases (we do not distinguish between modifiers and complements) and a head. Yet, in contrast to Eckle-Kohler (1999) and other CFG approaches, the embedded chunks/phrases do not have to be re-analysed for every query but can be accessed as structural mark-up (annotated in one of the previous steps). This makes it easier and faster to query optional elements, optional coordination and the special elements/chunks described above (see Table 1 and Table 2 for details). Recursion, in this context, is simply the inclusion of chunks/phrases of the same category that have been built in previous steps.

Chunks/phrases built in intermediate steps need not be the most complex structure possible, but are annotated to serve as input for further steps. Whenever a larger structure is found, the Perl scripts delete the smaller structure in favour of the new and more complex one. If the most complex structure of a phrase is not found, because of tagging errors or lacks in the grammar rules or the lexicon, the intermediate structures remain, and may, nevertheless, contribute to future extraction.

5 Grammar details

In the primary step special chunks/phrases are annotated along with features specifying their characteristics. These chunks are identified using one or more of the following criteria: (i) PoS-tags (multiword proper nouns), (ii) certain makers in the text itself such as brackets or quotes (terminology, multiword units), (iii) a lexicon in the form of word lists or subcategorisation information (temporal nouns, measure nouns, complex adjectival phrases), (iv) morpho-syntax (invariant or verbally derived adjectives). The features annotated with adjectival phrases and noun phrases are listed in Table 1 and Table 2.

Annotation features of adjectival chunks/phrases		
feature	description	examples
invar	invariant adjective	*Berliner Justiz; fünfziger Jahre*
vder	verbally derived adjective	*Beginnender Aufschwung; überzeugende Antworten*
meas	AC/AP embedding a measure noun	*Hektar groß; Meter langen*
norm	regular AC/AP	
quot	AC/AP in quotes	*´´falsche''; ´´allzu plötzlich''*
pp	AP embedding a PP or year date	*von seiner Frau geborgten; schon vor Jahren entdecktes*

Table 1: annotation features of adjectival chunks/phrases

Annotation features of noun chunks/phrases		
feature	description	examples
ne	proper noun	*Walter Holm; Mecklenburg-Vorpommern*
time	temporal noun	*Mai, Feierabend*
year	year date	*1999*
meas	measure noun	*Dollar Strafe; Hektar Pachtland*
news	news agencies	*LONDON (afp / rtr / AP)*
address	street name and number	*Musikantenweg 14; Museumsgasse 1*
tel	telephone numbers	*Tel. 23 33 25 oder 23 15 47*
trunc	nouns with truncs	*Schadenersatz- oder Schmerzensgeldanspruch; Verantwortungs-, Solidaritäts-, Gerechtigkeits-, Gleichheits- und Freiheitsdenken*
quot	NC/NP in quotes	*´´Autonome Organisation''; ´´United Democrats''*
brac	NC/NP in brackets	*(LKA); (Framingham Heart Study)*

Table 2: annotation features of noun chunks/phrases

Beside of these lexical/structural features we have also annotated (partially) disambiguated agreement features and the head lemma. As in complex grammars such as LFG or HPSG, the characteristics of the head project to the chunk/phrase. Similarly, the characteristics of intermediate chunks project to larger chunks/phrases with the same head. This projection of features is made possible by the use of Perl scripts that copy annotated features from smaller to larger structures, modifying them where necessary.

For instance, agreement disambiguation is performed in every intermediate step (where possible). The partially disambiguated agreement features are annotated along with the chunk/phrase. Later disambiguation steps use the annotated agreement features of relevant chunks/phrases (e.g., in the case of noun phrases, we use the agreement morphology of the determiner, the embedded adjectival phrases, and the noun chunk containing the head). Thus, (partial) disambiguation for a certain set of words does not have to be repeated.
A special method was chosen to check the agreement of multiple APs embedded in a NP. Every AP is annotated with a feature specifying its suffix (more precisely, the last letter of the adjective head, which is extracted by a Perl script in the first pass). All APs sharing the same suffix are assumed to agree in their morpho-syntactic features as well.[3] APs whose heads are invariant adjectives are ignored.

The chunks/phrases are built using relatively simple queries. A noun phrase, e.g., consists of a noun or year date as head and only obligatory element, which may have truncated elements (cf. Table 2). In pre-head position there are optionally a determiner, a cardinal number and a (theoretically) unlimited number of adjectival phrases. Cardinal numbers and adjectival phrases may be coordinated. Post-head there are optionally an unlimited number of genitive NPs, plus a noun chunk in quotes and/or in brackets:

das alkoholfreie Bier "Kelts"
the nonalcoholic beer "Kelts"

[3] The method was adopted from the German Gramotron grammar (cf. Schulte 2000).

den Namen "Werner-Herr-Haus"
the name "Werner-Herr-Haus"

das Quartett "Itchy Fingers"
the quartet "Itchy Fingers"

der Telefonnummer 602-316 (Herrn Borns)
the telephone number 602-316 (of Mister Born)

The pre-head as well as the post-head elements have to occur in the given order, yet, they do not depend on their predecessors. Table 3 gives a graphical overview of the noun phrase structure we use.

Elements of noun chunks/phrases				
pre-head		head	post-head	
optional	Determiner CARD* adjectival phrases* truncs*	noun proper noun	optional	genitive NC/NP proper noun NC in brackets NC in quotes
		year dates		
		Substitutive pronouns		
*optional coordination				

Table 3: elements of noun chunks/phrases

An adjectival phrase also has only one obligatory element, its head, which is an attributive adjective. Optional pre-head elements are adverbial phrases, a modifying particle (*allzu [großer]* (far too [big])*; zu [hohen]* (too [high])), and, in the case of verbally derived adjectives or adjectives subcategorising PPs, a prepositional phrase or a year date (see Table 4). A prepositional phrase simply consists of a preposition as its head and an obligatory noun phrase (which may optionally be coordinated)

Elements of adjectival chunks/phrases		
pre-head		Head
optional	adverbial phrase modifying particle	Adjective
optional	prepositional phrase year date	verbally derived adjective or adjectives subcategorising PPs

Table 4: elements of adjectival chunks/phrases

6 Evaluation

For evaluation purposes, we applied our parser to a 40-million-word newspaper corpus[4]. The corpus was preprocessed and part-of-speech tagged with standard tools (cf. Schmid 1995 and Schiller 1996). In order to reduce memory consumption during the parsing process, the corpus was automatically split into slices of approximately 500,000 tokens. Each slice was encoded as a separate corpus on which the parser was run. The structural annotations of all slices were then recombined and annotated in the original corpus.
The parsing process[5] took 13.5 hours on a standard 933 MHz Pentium III notebook with 128 MBytes of RAM. This amounts to an average speed of 3 million words per hour. Processing speed varied across different slices, with some slices taking almost twice as long as the "fastest" ones.

For a first quality assessment, we selected 100 sentences at random (excluding incomplete sentences and sentences containing spelling mistakes) and manually annotated noun phrases according to the extended chunk concept introduced in section 5. Unlike other manual annotation tasks, there is little ambiguity in the assignment of noun phrase boundaries when PP attachment is excluded, and agreement between the authors was easily reached.

[4] The *Frankfurter Rundschau* (FR) corpus from the ECI Multilingual CD-ROM I.
[5] Including the splitting and recombination steps, but *not* including preprocessing and part-of-speech tagging.

We did not evaluate the agreement features assigned to the phrases because they are only partially disambiguated and there is no guesswork on the side of the parser (any errors in case assignment etc. are due either to the morphological analysis or to tagging errors). For the same reason, other phrase types (such as PPs or APs) were not evaluated. PPs, for instance, can easily be identified by the parser once the corresponding NP has been found.

In the 100 test sentences, we found 477 noun phrases. Automatic parsing yielded 487 NPs, of which 440 were true positives. This corresponds to a *precision* of 90% (i.e. 90% of the NPs identified by the parser were correct) and a *recall* of 92% (i.e. 92% of the manually annotated NPs were also found by parser).

Looking at the results in detail, two major factors account for the number of false positives and NPs that were not found: tagging errors (e.g., the colon *:* tagged as a noun) and proper nouns that were not correctly identified (e.g., *Thomas Doll*, where *Doll* was erroneously tagged as an adjective). If we correct such errors (which can easily be done using lists of proper nouns etc.), the number of false positives drops from 47 to 13, and we obtain precision and recall values of 97% and 98%, respectively.

Another four of the eight noun phrases that our parser still cannot identify contain adjectives with PP complements. Feeding adjective subcategorisation frames extracted from the annotated corpus back into the parsing process will thus further improve the results.

7 Extraction

As mentioned above macro queries may not only be used for annotation but also for extraction. For this purpose they build on the results of the annotation process, searching on structural mark-up that they have produced before. In this case, the annotated features are an important knowledge source making morpho-syntactic information and characteristics of the chunks/phrases easily accessible. The extraction process itself can be automatic or semi-automatic depending on the query, i.e., the results of the queries may need manual checking before they can be used for the different purposes (e.g., lexicographic or linguistic).

Interesting for the extraction process are, e.g., chunks/phrases enclosed in brackets or quotes. These "structural" markers are relatively secure signs of elements belonging together. Thus, the elements enclosed can give hints regarding subcategorisation information of various kinds, but also with respect to multiword units, idiomatic expressions and collocations. Multiword units, in particular multiword proper nouns, sometimes occur in brackets or quotes, where they can be assembled securely. The same holds for abbreviations of terms, which often occur in brackets preceded by the respective term.

"Teenage Mutant Hero Turtles"
(FC Italia Frankfurt)
„Club Marienthaler Carnevalisten"
„Rocky Horror Picture Show"
„Johann Strauß Ensemble Frankfurt"
(Sprecher Wolf Hardy Pulina)
„Arbeiter Samariter Bund"
Technische Überwachung Hessen (TÜH)
Deutscher Aktienindex (Dax)
Daimler-Benz Inter Services (Debis)
Stickstoffdioxyd (NO2)

The annotation of structural mark-up can also help to model sentence positions in which certain elements occur without having to parse the whole sentence. Thus, the information these positions include can be accessed, even if the parse of the whole sentence is not successful. In the first position of German main clauses, for example, which is referred to as *Vorfeld* within the framework of the topological field model (cf. Wöllstein-Leisten et al. 1997), only one constituent may occur. Adverbs following NPs in this position can, consequently, be supposed to belong to the class of post-noun modifiers.

*Das "modernistische" Konzept **hingegen** lebt ...*
The „modernistic" concept however lives …

*Er **selbst** berichtet ...*
He himself reports ...

*Die Rationalität **alleine** ist ...*
The rationality alone is ...

*Herrn Frank **persönlich** wünsche ...*
Mister Frank personally wishes ...

*Das Volk **indessen** lässt ...*
The people meanwhile let ....

It is also possible to overgenerate certain structures, annotating them only temporarily in the corpus. If they are not embedded in a larger construction or prove to be correct in another way, they may be deleted again, otherwise they remain annotated in the corpus. Due to this possibility, structures can be annotated that would need subcategorisation information not available in the lexicon. These structures can than be queried and taken as evidence for certain subcategorisation frames. An example are adjectives subcategorising PPs that can build complex adjectival phrases with the respective PP. In this case, we overgenerate APs allowing all adjectives intermediately to build complex APs with preceding PPs or year dates. The APs are deleted again, unless they are embedded in a NP after the last annotation step, i.e., they are preceded by a cardinal number or determiner belonging to the head noun of the NP.

*Einem **auf die Betreuung Aidskranker spezialisierten** Sozialarbeiter*
A on the care of people suffering from aids specialised social worker
"A social worker specialising in the care of people suffering from aids"

*Der **für chinesische Verhältnisse kleinen** 20 000 Einwohner zählenden Stadt*
The for Chinese standards small 20 000 inhabitant having city
"The city with 20 000 inhabitants, a small city by Chinese standards"

*Die **dafür erforderlichen** 300 000 Mark*
The for this needed 300 000 Marks
"The 300 000 Marks needed for this"

*Die **für eine Aufrufung des Rates notwendigen** 60 Abgeordneten*
The for a summoning of the council necessary 60 delegates
"The 60 delegates necessary to summon the council"

8 Acknowledgments

Part of our work was done within the framework of the DEREKO project. The DEREKO (**De**utsches **Re**ferenz**ko**rpus) project is a joint project of the Institut für deutsche Sprache (IDS) in Mannheim, the Institute for Natural Language Processing (IMS) in Stuttgart, and the Seminar für Sprachwissenschaft (SfS) in Tübingen. The project is funded by the Ministry of Science, Research and the Arts of the State of Baden-Württemberg. The goal of the project is to improve the infrastructure for text-based linguistic research and development by making accessible a large well-balanced German text corpus. This corpus is intended as a source for linguistic and lexicographic information. The target group for the resulting infrastructure are lexicographers, dictionary publishers, manufactures of terminological databases and ontologies as well as linguists.

9 References

Abney S 1991 Parsing by chunk. In Berwick R, Abney S, Tenny C (eds), *Principle-based parsing*. Dordrecht, Kluwer Academic Publishers.

Abney S 1999 Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.

Christ O 1994 A modular and flexible architecture for an integrated corpus query system. In *Papers in Computational Lexicography COMPLEX '94.* Budapest, Hungary, pp 22-32.

Christ O, Schulze B M, Hofmann A, König E 1991 *Corpus Query Processor (CQP). User's Manual*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Germany.

Eckle-Kohler J 1999 *Linguistisches Wissen zur automatischen Lexikon-Akquisition aus deutschen Textcorpora.* Berlin, Logos Verlag.

Schulte im Walde S 2000 The German statistical grammar model: development, training and linguistic exploitation. In *Arbeitspapiere des Sonderforschungsbereichs 340 Linguistic Theory and the Foundation of Computational Linguistics 162*. Institut für Maschinelle Srpachverarbeitung, Universität Stuttgart, Germany.

Wöllstein-Leisten A, Heilmann A, Stepan P, Vikner S 1997 *Deutsche Satzstruktur. Grundlagen der syntaktischen Analyse*. Staufenberg Verlag, Tübingen.