Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-91-18

A Diagnostic Tool for German Syntax

John Nerbonne, Klaus Netter,

Abdel Kader Diagne, Ludwig Dickmann, Judith Klein

July 1991

Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Philips, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❏ Intelligent Engineering Systems
- ❏ Intelligent User Interfaces
- ❏ Intelligent Communication Networks
- ❏ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

A Diagnostic Tool for German Syntax

John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann and Judith Klein

DFKI-RR-91-18

This paper will also be published in the Proceedings of the 'Natural Language Processing Systems Evaluation Workshop' held in connection with the 29th Annual Meeting of the ACL, Berkeley Ca., June 1991. The proceedings will appear as a Rome Laboratory Technical Report.

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9002 0).

A Diagnostic Tool for German Syntax *

John Nerbonne, Klaus Netter
Abdel Kader Diagne, Judith Klein[†] and
Ludwig Dickmann[‡]

[†] Deutsches Forschungszentrum für Künstliche Intelligenz, GmbH
Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, FRG
phone: (+49 681) 302-5300
e-mail: nerbonne@dfki.uni-sb.de

and

[‡]Institut für Computerlinguistik, Universität des Saarlandes
Im Stadtwald, D-6600 Saarbrücken 11, FRG

Abstract

In this paper we describe an effort to construct a catalogue of syntactic data, exemplifying the major syntactic patterns of German. The purpose of the corpus is to support the diagnosis of errors in the syntactic components of natural language processing (NLP) systems. Two secondary aims are the evaluation of NLP systems components and the support of theoretical and empirical work on German syntax.

The data consist of artificially and systematically constructed expressions, including also negative (ungrammatical) examples. The data are organized into a relational data base and annotated with some basic information about the phenomena illustrated and the internal structure of the sample sentences. The organization of the data supports selected systematic testing of specific areas of syntax, but also serves the purpose of a linguistic data base.

The paper first gives some general motivation for the necessity of syntactic precision in some areas of NLP and discusses the potential contribution of a syntactic data base to the field of component evaluation. The second part of the paper describes the set up and control methods applied in the construction of the sentence suite and annotations to the examples. We illustrate the approach with the example of verbal government. The section also contains a description of the abstract data model, the design of the data base and the query language used to access the data. The final sections compare our work to existing approaches and sketch some future extensions.

We invite other research groups to participate in our effort, so that the diagnostics tool can eventually become public domain.

*This work was supported by a research grant, ITW 9002 0, from the German Bundesministerium für Forschung und Technologie to the DFKI project DISCO and by IBM Germany through the project LILOG-SB conducted at the University of Saarbrücken.

Contents

1 Introduction

This paper describes an effort to construct a catalogue of syntactic data which is intended eventually to exemplify the major syntactic patterns of the German language. Our purpose in developing the catalogue and related facilities is to obtain an empirical basis for diagnosing errors in natural language processing systems analyzing German syntax, but the catalogue may also be of interest to theoretical syntacticians and to researchers in speech and related areas. The data collection differs from most related enterprises in two respects: (i) the material consists of systematically and artificially constructed sentences rather than naturally occurring text, and (ii) the material is annotated with information about the syntactic phenomena illustrated, which goes beyond tagging parts of speech. The catalogue currently treats verb government, (including reflexive verbs and verbal prefixation) and coordination.

The data consists of linguistic expressions (mostly short sentences designed to exemplify one syntactic phenomenon) together with annotations describing selected syntactic properties of the expression. The annotations of the linguistic material serve (i) to classify construction types in order to allow selected systematic testing of specific areas of syntax, e.g., coordination; and (ii) to provide a linguistic knowledge base supporting the research and development of natural language processing (NLP) systems. Besides classificatory information, the annotations contain information about the precise structure of the sentence such as the position of the finite verb and the positions of other phrases.

In order to probe the accuracy of NLP systems, especially the detection of unwanted overgeneration, the test material includes not only genuine sentences, but also some syntactically ill-formed strings.

The syntactic material, together with its annotations is being organized into a relational database in order to ease access, maintain consistency, and allow variable logical views of the data. The database system is in the public domain and is (mostly) independently supported.

Our intent is to make public this work—both the test material and the database of annotations. We plan to share this work first with selected contributing partners, and later with the general research and development community.

2 Goals of a Diagnostics Tool

Our goal in collecting and annotating syntactic material is to develop a diagnostic tool for natural language processing systems, but we believe the material may be of interest to other researchers in natural language, particularly syntactic theoreticians. Finally, although this is not an evaluation tool by itself, our work points to possiblities for evaluating systems of syntactic analysis by allowing the systematic verification of claims about, and investigation of, the coverage and precision of systems.

2.1 Natural Language Processing

There is general consensus, both in theoretical computational linguistics and in practical, industrially sponsored research in natural language processing, that systems for syntactic analysis (parsing, recognition and classification) are possible and valuable. The applications of syntactic analysis currently under investigation include grammar and style checking; machine translation; natural language unterstanding (particularly interfaces to databases, expert systems, and other software systems); information retrieval; speech synthesis; and speech recognition. The potential impact of syntactic analysis technology is technically and financially profound.

But if we are to realize the full benefits of syntactic analysis, then we must ensure that correct analyses are provided. The development of a diagnostic tool serves just this purpose—pointing out where analyses are correct, and where incorrect. There are, of course, other measures of quality which apply to natural language software, e.g., general software standards. Systems which perform syntactic analysis are naturally subject to the same general standards of software quality that are imposed throughout the software engineering field, e.g., efficiency, modularity, modifiability, compatibility, and ease of installation and maintenance. Special-purpose systems may be subject to further standards; e.g., interface software is generally required to have clear and intuitive boundaries (transparency). Compared to such general software standards, correctness of syntactic analysis is an orthogonal criterion, though for many applications, an overriding one. Attending exclusively to general software standards means risking incorrectness—whether this be incorrectness of matrix multiplication in a linear algebra package or misanalyses in a natural language parser. The ultimate costs of such misanalysis depend, of course, on the particular application, but these costs may easily outweigh the benefits of the system deployed.

The importance of precision in syntactic analysis is occasionally disputed. It is pointed out, for example, that humans make speech errors (and typos), and that natural language understanding systems will have to be sufficiently robust to deal with these. Here, it is claimed, less precise systems may even have an advantage over more exact, and hence "brittle" competitors. What is correct about this point is that systems should be able to deal with ill-formed input. What is questionable is the suggestion that one deal with it by relaxing syntactic or other constraints *generally* (although it might be quite reasonable to use constraint relaxation where no exact analysis may be found—as a processing strategy).

The problem with general constraint relaxation is that it inevitably involves not only providing analyses for ill-formed input (as intended), but also providing additional incorrect analyses for well-formed input—"spurious ambiguity". To see this, consider agreement, probably a good candidate for a less important "detail" of syntax which might safely be ignored. For example, it might be argued that sentence (1) below ought to be regarded as syntactically acceptable, since it's clear enough what's intended:

(1) *Liste alle Sekretärinnen, die einen PC benutzt*
List all secretaries who uses a PC

Syntactically tolerant systems would accept this sentence, but they would then have no way of distinguishing correct and incorrect parses of sentences such as (2), which are distinguished only by agreement:

(2) *Liste jede Sekretärin in Finanzabteilungen, die einen PC benutzt*
List every secretary in finance departments who uses a PC

The relative clause *die einen PC benutzt* can of course only be understood as modifying *jede Sekretärin* (the only NP with which it agrees), but a system which ignored agreement information would have no way of eliminating the parse in which the relative clause is construed as modifying *Finanzabteilungen*.

Furthermore, even if we accepted the argument that some applications may ignore syntactic accuracy, we are still faced with the applications at the other end of the spectrum of syntactic sensitivity, i.e., applications where syntactic accuracy is essential. Applications of this sort are found where the microstructure of the text plays an important role, e.g., grammar or style checking, and generally the entire area of NL generation: clearly, nobody wants a system which over-generates in synthesis. Similarily it is hard to find any advantage for underconstrained systems in applications such as speech understanding, where the whole point of using syntactic information is to reduce the number of hypotheses—a goal served only by maximally constrained systems.

We therefore believe that syntactic precision is indispensable for some applications and valuable even in applications in which ill-formed input may be expected.

The diagnostic tool assesses correctness of syntactic analysis—it supports the recognition of bugs in the linguistic analysis. This in turn provides both a means of assessing the effects of proposed changes in syntactic analysis as well as a means of tracking progress in system coverage over time. Neither of these deriative tasks is realistically feasible without the aid of an automated tool. Humans may spot individual errors when attending propitiously, but we're poor at systematic checks and comparisons, especially in large systems created by groups over relatively long periods of time.

2.2 Linguistic Research

This is an appropriate point at which to acknowledge our own debt to descriptive and theoretical linguistics, from which our primary data—the German sentences themselves—have been gathered. We expect to reciprocate, i.e., we expect that descriptive linguistics and even linguistic theory may benefit from the data collection effort we have undertaken. These benefits may take different forms: first, we have begun gathering the data in a single place; second, we are organizing it into a database in a fairly general way, i.e. with relatively little theoretical prejudice, so that variable perspectives on the data are enabled; third, in addition to relatively crude data analysis routinely provided in linguistic data collections—which seldom extends beyond marking ill-formedness/well-formedness, we have provided further fundamental data annotations. Fourth, and most intriguingly , the time may not be distant when linguistic hypotheses may be tested directly on the computer. Many contemporary computational systems for natural language syntax are based on ideas of current interest in theoretical linguistics as well, and there is interest in general machinery for implementing syntactic analysis for wide varieties of linguistic theories. At that point, the use of diagnostic tools will be of immediate interest in linguistic research as well.

In sketching these potential benefits of the general data collection and analysis effort we have begun, it should be clear that we don't intend to speak only to linguists emploring "corpus-based" methodologies: our information includes facts about the ill-formedness of strings as well as rudimentary data analysis. This will become clearer below.

2.3 Toward Evaluation

The catalogue of syntactic material we have collated is intended for deployment in diagnosis—the recognition and characterization of problems in syntactic analysis. This is a task different from general system evaluation, which in most cases will judge the performance of a system relative to the achievement of a goal which is set by an application. Even if we limit evaluation to the performance of the syntactic component of a system, there are still some differences which have to kept in mind.

The contrast between diagnosis and evaluation can be appreciated if one considers the case of applying our diagnostic tool to two different systems. In virtually every case, the result we obtain will show that neither system is perfect (nor perfectly incorrect), and that neither one analyzes exactly a subset of the constructions of the other. Suppose, for the sake of illustration, that one system is superior in treating long-distance (multi-clausal) dependencies, while the other is better at simple clause structure, but that the performance of the two systems is otherwise the same. The diagnosis is complete, but the evaluation still needs to determine the relative importance of the areas in which coverage diverged.[1] If matters were always as simple as in this illustration, we might appeal to a consensus of informed opinion, which would in this case certainly regard the treatment of simple clause structure as more important than that of long-distance dependencies—and would therefore evaluate the systems accordingly. But matters need not and normally are not so simple at all. There simply is not a consensus of informed opinion about the relative importance of various areas of grammatical coverage.

Some crucial information that is lacking from our catalogue of syntactic material is information about relative frequency of occurrence. If this information could be obtained and added to the database, then it should be possible to develop an evaluation system of sorts from our diagnosis system.[2]

[1] Strictly speaking, this is not necessary; we could evaluate all such cases as equally proficient, but (i) the results of such "evaluation" would be too coarse to be of much use; and (ii) this simply goes against good sense. Some areas of grammatical coverage simply are more important than others. See the example in text, where simple clause structure is certainly more important long-distance (multi-clausal) dependency.

[2] But it is not clear that this is the best way to go about developing an evaluation system. For example, we are not making any effort to keep some of the material secret, as speech evaluation systems routinely do in order to prevent a bias toward test material.

3 The Diagnosis Facility

We include here a brief description of the diagnostic facility; more detailed documentation, especially for the various areas of coverage of the syntactic catalogue, is currently under preparation.

3.1 Sentence Suite

As noted in the introduction, our material consists of sentences we have carefully constructed to illustrate syntactic phenomena; we have not attempted to collect examples from naturally occurring text. Several considerations weighed in favor of using the the artificially constructed data:

- since the aims are error detection, support of system development, and evaluation of systematic coverage, we need optimal control over the test data. Clearly, it is easier to construct data than to collect it naturally when we have to examine (i) a systematic range of phenomena or (ii) very specific combinations of phenomena.
- we wished to include negative (ill-formedness) data in order to test more precisely (cf. discussion in Section 2.1 on "spurious ambiguity" and also on the needs of generation). Negative data is not available naturally.
- we wished to keep the diagnostic facility small in vocabulary. This is desirable if we are to diagnose errors in a range of systems. The vocabulary used in the diagnostic tool must either (i) be found in the system already, or (ii) be added to it easily. But then the vocabulary must be limited.
- we wished to exploit existing collections of data in descriptive and theoretical linguistics. These are virtually all constructed examples, not naturally occurring text.
- data construction in linguistics is analogous to the control in experimental fields—it allows the testing of maximally precise hypotheses.

We have no objection to including naturally occurring data in the catalogue, subject to the restrictions above (especially constraining the size of the facility).

The vocabulary for the test suite has been taken from the domain of personnel management wherever possible. We chose this domain because it is popular in natural language processing, both as a textbook example and as an industrial test case. The domain of personnel management would also be useful in case we are to diagnose errors in semantics as well as syntax (which we are not attempting to do at present, but which is an interesting prospect for the future). It presents a reasonably constrained and accessible semantic domain. Where no suitable vocabulary from the domain of personnel management presented itself, we have extended the vocabulary in *ad hoc* ways.

The suite of test sentences is being collated by various contributors, each specializing in a single area of coverage, e.g. verb government, coordination, or NP constructions. Because of the range of syntactic material which is eventually to be included, it is difficult to draw precise guidelines about the sentences.

Still, several factors have been borne in mind while constructing the syntactic examples.

- lexicon size (cf. above)
- adherence to the following standards: (somewhat) formal, conversational High German; i.e., we have avoided colloquialisms, literary peculiarties, and regional dialects.
- selected testing of negative examples. We have tried to keep the catalogue small, but not at the cost of using great ingenuity to create minimal sets of testing data, nor at the cost of introducing very unnatural examples into the test catalogue. We have not rigorously purged superfluous examples.
- minimization of irrelevant ambiguity (bearing in mind that it cannot be fully eliminated).
- attention to analytical problems. We have attempted to catalogue not only the constructions, but also the problems known to be difficult in their analysis.

We do not deceive ourselves about our chances for success with respect to the last point: our catalogue is doubtlessly incomplete in many respects, but most sorely in this one. We invite comment and contribution everywhere, but most especially in further cataloguing the known analytical problems in German syntax.

In stressing our intention to catalogue analytical problems as well as the basic range of syntactic construction types, we do not intend to suggest that we intend to gather a collection of "cute examples". We will gather cute examples, but these are relatively few in the general catalogue. Our primary goal will be a coverage of phenomena which is as comprehensive as feasible, even if this involves the rather tedious compilation of theoretically relatively well-explored and scientifically "uninteresting" constructions, such as the full paradigms illustrating determiner-adjective-noun agreement in German or the different types of verbal subcategorization. From our experience, it is above all the absence of systematic and comprehensive test-beds which hampers system development, rather than the lack of ingenious examples (which frustrate all systems in some way or other). Our goal is thus not primarily to show what systems cannot do, but to support the extension of what they can do.

3.2 Syntactic Annotations

In choosing which annotations about the sentences might be sensible, we have been guided by two considerations. First, the catalogue will be much more useful if examples from *selected* areas can be provided on demand. For example, it would be useful to be able to ask for examples of coordination involving ditransitive verbs—as opposed to simply coordination (an area of coverage). This means that we need to provide annotations about which area of coverage a given sentence (or ill-formed string) is intended to illustrate. With regard to these annotations, we have merely attempted to use standard (traditional) linguistic terminology.

Second, we can exploit some annotations to check further on precision of analysis. This is the purpose of annotations such as:

- well-formed vs. ill-formed
- position of finite matrix verb
- position of NP's
- position of PP's

So, in a sentence such as (3), the following database values are encoded:

(3) *Der Student bittet den Manager um den Vertrag.*
the student asks the manager for the contract

*/OK	OK
finite matrix verb	3
position of NP's	1-2, 4-5, 7-8
position of PP's	6-8

In selecting these properties as worthy of annotation, we were motivated primarily by a wish to focus on properties about which there would be little theoretical dispute, which would be relatively easy to test, and which would still provide a reasonable reflection of a system's accuracy.

3.3 An Example: Verbal Government

One of the phenomena which the data collection already covers is the area of verbal government, i.e., verbal subcategorization frames. The aim was to compile a comprehensive list of *combinations of obligatory arguments* of verbs, forming the basis of different sentence patterns in German. We ignore both adjuncts and optional arguments in restricting ourselves to obligatory arguments, which can be tested by an operationalizable criterion, a specific sort of right extraposition:

(4) *Er hat gegessen, und zwar Bohnen.*
he has eaten, namely beans.

(5) **Er hat verzehrt, und zwar Bohnen.*
he has consumed, namely beans

(6) **Er hat das Buch gelegt, und zwar auf den Tisch.*
he has put the book, namely on the table

(7) *Er hat Maria geküßt, und zwar auf die Wange.*
he has kissed Mary, namely on the cheek

We attempted to find instances of all possible combinations of nominal, prepositional, sentential, but also adjectival complements.[3] Clearly, we could not immediately cover the entire field in full depth, so that we decided to adopt a breadth first strategy, e.g., we ignored the more finegrained distinctions to be made in the area of infinitival complementation or expletive complements. The description in these areas will be elaborated at later stages.

The result of the collection is a list of about 90 combinations which are exemplified in about 300 sample sentences.

The sentences illustrate

- combinations of nominal, prepositional and adjectival arguments, viz.,
  - nominal arguments only:

    (8) *Der Manager gibt dem Studenten den Computer.*
    the manager gives the student the computer

  - nominal and prepositional arguments with semantically empty (9) or non-empty prepositions (10):

    (9) *Der Vorschlag bringt den Studenten auf den Lösungsweg.*
    the suggestion takes the student to the solution

[3] At the basis of our list were collections to be found in the literature, such as [2], [5], [6], [7], [9] and [12]. We are also grateful to Stefanie Schachtl, Siemens Munich, who provided us with some of her material.

(10) *Der Manager vermutet den Studenten in dem Saal.*
the manager assumes the student in the hall

– nominal and adjectival (or predicative) complements

(11) *Der Manager wird krank.*
the manager becomes ill

- nominal arguments combined with finite (subordinate) clauses, introduced by the complementizers *daß* (12), *ob* (13) or some wh-element (14):

(12) *Daß der Student kommt, stimmt.*
that the student comes, is-correct

(13) *Dem Manager entfällt, ob der Student kommt.*
it escapes the manager, whether the student comes

(14) *Der Manager fragt, wer kommt.*
the manager asks who comes

- nominal arguments in combination with infinitival complements, illustrating bare infinitives (15) and *zu*-infinitives (16):

(15) *Der Manager hört den Studenten kommen.*
the manager hears the student come

(16) *Der Manager behauptet, den Studenten zu kennen.*
the manager claims to know the student

- examples involving some of the combination above in connection with expletive or correlative prepositional pronouns or expletive *es*:

(17) *Der Vorschlag dient dazu, den Plan zu erklären.*
the proposal serves (to-it) to explain the plan

(18) *Der Manager achtet darauf, ob der Student kommt.*
the manager checks (on-it) whether the student comes

(19) *Es gelingt dem Studenten, zu kommen.*
it succeeds to the student, to escape
"The student succeeds in escaping"

(20) *Der Manager hält es für notwendig, zu kommen.*
the manager considers it (for) necessary to come

Since we are interested only in verbal government here, we tried to keep as many other parameters as possible carefully under control: as already mentioned, the noun phrases in the sample sentences are built from a limited vocabulary. All noun phrase and prepositional complements have a definite determiner. In the case of prepositional phrases the fusion of preposition and determiner (*in dem*

→ *im*) is avoided. Since German has relatively free word order, the different complements have to be identified by their case marking in most cases—as a consequence, morphological ambiguities of case (e.g. between feminine or neuter nominative and accusative) were excluded. The matrix and subordinate clauses all have only one verbal head (i.e., they do not have any auxiliary or modal verbs), whose morphological form is the third person, singular, present, indicative form if possible. The sentences do not contain any additional irrelevant modifiers, adjuncts or particles. The word order of the sample sentences is meant to illustrate the "un-marked" order, although this should not play an important role, since the complements are uniquely case marked, as mentioned.

Every combination of complements is illustrated by at least one example. In addition, each sentence is paired with a set of ill-formed sentences, which illustrate three types of errors relevant for verbal government:

- an obligatory argument is missing;
- there is one argument too many;
- one of the arguments has the wrong form.

The material is organized in a relational database, such that queries can ask either for a description or classification of a sentence or for sentences matching combinations of descriptive parameters.

In describing the argument structure of the sentences we chose a vocabulary which is of course not theory neutral, but which at least can be expected to meet common agreement. We tried to avoid theory-specific notions such as *subject* or *direct object*, and identified the complements on the basis of morphological case marking, prepositions, complementizers and/or the morphology of the verb. Obviously, this vocabulary cannot exhaustively characterize the properties of individual complements. For example, with those few verbs which subcategorizes for two accusative NPs it is quite unlikely that they both NPs behave in the same way with respect to passivization. Similarily, a nominative complement ("subject") may have different propertives depending on the verb being un-accusative or un-ergative. However, we think that distinctions of this kind should be dealt with seperately in data sets on e.g. passivization, ergativity, etc.

3.4 Database

3.4.1 Abstract Data Model

In addition to the relatively straightforward properties of sentences noted above (Section 3.2), we also model the more complex classificatory information in the catalogue.

According to the **Entity-Relationship** (ER) terminology (cf. [3]), we can identify two entity types and one relationship type which are specified as follows:

1. SENTENCE, an entity type, the major concept of the data model. An entity of this type includes a description of the main verb's valency (i.e., the number of arguments the main matrix verb governs and their description), a sentence which exemplifies the given properties, and information on its wellformedness. Each entity has a unique identifier, a key attribute which facilitates queries for description or classification of a sentence. (Given the present limited range of data and the underlying area (verb government), the attributes argument-description and fin-matrix-verb could almost be used to identify a sentence entity uniquely, because there is only one representative from most valency types in SENTENCE. But some types are represented more than once.)
2. CATEGORY, an entity type. Each entity of this type (e.g., NP, finite_matrix_verb) represents a category which appears in a related sentence.
3. APPEARS_IN, a M:N relationship type[4] between CATEGORY and SENTENCE. Both CATEGORY and SENTENCE participations in the relation are total. APPEARS_IN has additional attributes specifying the position of a given category in a related sentence and its lexical form.

The following figure illustrates the conceptual model of the database described above. It covers the area of verbal government and can be easily extended.

[4] M:N relation (many to many relation): a sentence entity may be related to (i.e. may include) numerous category entities, and a category entity may appear in numerous sentence entities.

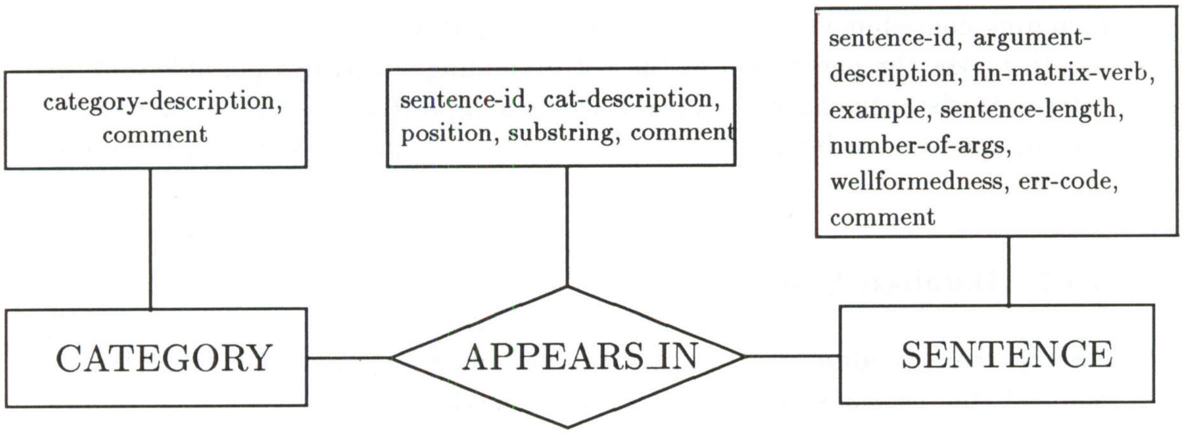
category-description, comment
sentence-id, cat-description, position, substring, comment
sentence-id, argument-description, fin-matrix-verb, example, sentence-length, number-of-args, wellformedness, err-code, comment
CATEGORY
APPEARS_IN
SENTENCE

Figure 1: The ER schema diagram for the database described above.

The following example shows database entries for a given sentence.

(21) *Der Vorschlag bringt den Studenten darauf, daß der Plan falsch ist.*
the suggestion takes the student to-it, that the plan is wrong

SENTENCE

s-id	**arg-description**	**m-m-v**	**ex**	**sl**	**na**	**wf**	**err**	**com**
1012	nom_acc_cor_sc_dass	bringen	(s1)	11	4	1	0	-

(s-id = sentence-id, m-m-v = fin-matrix-verb, ex = example, sl = sentence length, na = number of arguments, wf = wellformedness, err = error code, com = comment)

CATEGORY

category-description	**comment**
cor	correlate
fin-matrix-verb	
NP	
sc-comp	subordinate clause

APPEARS_IN

sentence-id	**category-description**	**pos-from**	**pos-to**	**substring**
1012	cor	6	6	darauf
1012	fin-matrix-verb	3	3	bringt
1012	NP	1	2	der Vorschlag
1012	NP	4	5	den Studenten
1012	sc-comp	7	11	daß der Plan falsch ist.

A new database entry for a given sentence must include values for the attributes arg-description, fin-matrix-verb, example, wellformedness, category-description, pos-from, and pos-to. For ill-formed sentences the error code and additional comments should be given. All other attributes can be inserted through some triggers including consistency checks. Splitting the position attribute into pos-from and pos-to makes the generation of the corresponding substrings possible and facilitates a consistency check (e.g., pos-from must a positive integer number less than pos-to, pos-to must be greater than pos-from and equal or less than the sentence length.).

3.4.2 Database System

The database is administered in the programming language **awk** (cf. [1]). Some of the reasons which speak in favor of **awk** are:

- **awk** is in the public domain running under UNIX and should run in other environments; in particular, it runs on MS-DOS.
- Its ability to handle strings of characters as conveniently as most languages handle numbers makes it for our purposes more suitable than standard relational database systems; i.e., it allows more powerful data validation, increasing the availability of information with a minimal number of relations and attributes.

Compared to standard databases **awk** has a restricted area of application and does not provide fast access methods to information, but it is a good language for a developing a simple relational database in a number of cases. Additional resources and tools such as a report generator and a routine for consistency checking can be easily implemented.

The database includes a reduced **sql**-like query language. We use the database entries of the example given above to ask the following queries:

(i) retrieve all sentences which include a correlate and a subordinate clause beginning with *daß*.

```
query: retrieve sentence-id, example
       from sentence
       where match(arg-description, "cor") and match(arg-description, "sc-dass")

result: 1012   der Vorschlag . . . falsch ist.
```

(ii) retrieve the position and the lexical form of all NP's of sentence 1012.

```
query: retrieve cat-description, position, substring
       from sentence, appears_in
       where sentence-id = 1012 and category-description = "NP"

result: NP  1  2  der Vorschlag
        NP  4  5  den Studenten
```

The query language has been developed under SunOS using the utilities **lex** and **yacc**. **lex** is a lexical analyzer generator designed for processing of character input streams. **yacc**, a LALR(1) parser generator, is an ancronym for Yet Another Compiler Compiler. It provides a general tool for describing an input language to a computer programm.

3.5 Auxiliary Materials

The database of syntactic material is to be accompanied by a few auxiliary development tools. First, in order to support further development of the catalogue and database, it must be possible to obtain a list of words used (so that we minimize vocabulary size), and a list of differentiating concepts (so that categorization names may be accessed easily). Second, documentation must be available on each of the areas of syntactic coverage included. This is to cover (minimally) the delimitation of the area of coverage, the scheme of categorization, and the sources used to compile the catalogue.

Third, a small amount of auxiliary code may be supplied to support development of interfaces to parsers. This need not do more than dispatch sentences to the parser, and check for the correctness of results.

4 Comparison to Other Work

This appears to be the first attempt to construct a general diagnostic facility for German syntax, even if virtually every natural language processing group working on German has a small suite of sentences used for internal monitoring and debugging.

There have been several related efforts concerned with English syntax. Guida and Mauri [8] report on attempts to evaluate *system* performance for natural language processing systems (*n.b.*, not merely syntax) in which they attempt to finesse the issue of correctness (which we argue to be central) by measuring user satisfaction. We have attempted to address the issue of syntactic correctness head-on.

Hewlett-Packard Laboratories compiled a test suite of approximately $1,500$ sentences which it distributed at the *Public Forum on Evaluating Natural Language Systems* at the 1987 Meeting of the Assocation for Computational Linguistics [4]. That effort differed from the present one in that it tried to evaluate semantics and pragmatics, as well as syntax, and in that it consisted essentially of sentences without annotated properties. The sentences were not organized into a database.

Read et al. [11] advocate a "sourcebook" approach, in which fewer examples are submitted to much closer scrutiny. The closer scrutiny doesn't seem subject to automation, at least at present. Furthermore, their emphasis is on evaluating *systems* for natural language understanding, and the primary focus seems to be on domain modeling, conceptual analysis and inferential capabilities, not syntax. It is similar to the HP approach (and to ours) in employing primarily constructed examples, rather than naturally occurring ones.

The Natural Language group at Bolt, Beranek, and Newman Systems and Technologies Corporation circulated a corpus of approximately $3,200$ sample database queries formulated in English at the 1989 DARPA Workshop on Evaluating Natural Language [10]. The emphasis here, too, was on system (natural language understanding) performance, rather than specializations, but most of their examples seem to come from actual trial use of a natural language interface program, which gives their work added value.

The University of Pennsylvania's "Treebank" project (similar to a project of the same name at the University of Lancaster sponsored by IBM) has begun an effort to annotate naturally occurring text and speech, and to organize the annotations into a "Treebank". The annotations are phonetic, syntactic, semantic and pragmatic, and the intended scope is monumental. Since they wish to gather representative and varied data, they hope to collect and annotate approximately 10^8 words.

Finally, the Text-Encoding Initiative of the Association for Computational Linguistics is a loosely organized confederation of efforts concerned with the classification and annotation of various sorts of texts. Our work will be made available to this group.

5 Current State, Future Plans

5.1 Collaborations

We have contacted some research groups in the area of NLP and machine translation, which have shown interest in cooperating on the effort by submitting data sets in exchange for the use of the database. Among these are the Institut für angewandte Informationswissenschaft (IAI), Saarbrücken and a research and development group at Siemens, Munich.

5.2 Eventual Range of Syntax Catalogue

As mentioned, we regard our work only as a starting point which has to be complemented by contributions from other groups and individual experts. As to extensions of the database, we can only provide the roughest of lists here. We intend the list to be suggestive rather than definitive:

Syntax of the simple clause, including verbal government and *genera verbi* (passive, etc.), negation, word order, and adverbial modification, including temporal adverbials (duratives, frequentatives, and "frame" adverbials), locative, manner, and measure adverbials. Verb phrase complementation including argument sharing or inheritance (*auf Hans ist er stolz*), clause union, extraposition, modal and auxiliary verbs. Verbal complex, fixed verbal structures (*Funktionsverbgefüge*), separable prefix verbs, idioms and special constructions.

Noun phrase syntax, including determiner and numeral (and measure) system, relative clauses of various sorts (including preposed participial phrases), pre- and postnominal adjectival modification, noun phrase coordination, and plurals. Pronominal system and anaphora.

Prepositons and postpositions, cliticization, particles (e.g., *als, ja, je, denn*).

Questions, including long-distance (multi-clause) dependence. Imperative and subjunctive moods. Adjectival and nominal government, modification, and specification. Equative, comparative, and superlative constructions. Coordination and ellipsis.

References

[1] A.V. Aho, B.W. Kernighan and P.J. Weinberger: *The awk programming language.* Wokingham et al., Addison Wesley, 1988

[2] Peter Colliander: *Das Korrelat und die obligatorische Extraposition.* Kopenhagener Beiträge zur Germanistischen Linguistik. Sonderband 2. Kopenhagen, 1983.

[3] P. Chen: The entity-relationship model. Toward a unified view of data. *ACM Transactions on Database Systems.* No. 1, 1976.

[4] Daniel Flickinger, John Nerbonne, Ivan Sag, and Thomas Wasow: Towards evaluation of natural language processing systems. Technical report, Hewlett-Packard Laboratories, 1987.

[5] Ulrich Engel: Die deutschen Satzbaupläne. In *Wirkendes Wort 20,* pages 361–392, 1970.

[6] Bernhard Engelen: *Untersuchungen zu Satzbauplan und Wortfeld in der geschriebenen deutschen Sprache der Gegenwart.* Reihe I 3.3 Verblisten. München, 1975.

[7] Lutz Götze: *Valenzstrukturen deutscher Verben und Adjektive.* München, 1979.

[8] Giovanni Guida and Giancarlo Mauri: Evaluation of natural language processing systems: Issues and approaches. *Proceedings of the IEEE,* 74(7):1026–1035, 1986.

[9] Gerhard Helbig: *Wörterbuch zur Valenz und Distribution deutscher Verben.* Leipzig, 5th ed., 1980.

[10] Martha Palmer, Tim Finin, and Sharon M. Walter: Workshop on the evaluation of natural language processing systems. Technical Report RADC-TR-89-302, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, 1989.

[11] Walter Read, Alex Quilici, John Reeves, Michael Dyer, and Eva Baker: Evaluating natural language systems: A sourcebook approach. In *COLING '88,* pages 530–534, 1988.

[12] Monika Weisgerber: *Valenz und Kongruenzbeziehungen.* Frankfurt a. M., 1983.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen oder die aktuelle Liste von erhältlichen Publikationen können bezogen werden von der oben angegebenen Adresse.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of currently available publications can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-01
Franz Baader: Terminological Cycles in KL-ONE-based Knowledge Representation Languages
33 pages

RR-90-02
Hans-Jürgen Bürckert: A Resolution Principle for Clauses with Constraints
25 pages

RR-90-03
Andreas Dengel, Nelson M. Mattos: Integration of Document Representation, Processing and Management
18 pages

RR-90-04
Bernhard Hollunder, Werner Nutt: Subsumption Algorithms for Concept Languages
34 pages

RR-90-05
Franz Baader: A Formal Definition for the Expressive Power of Knowledge Representation Languages
22 pages

RR-90-06
Bernhard Hollunder: Hybrid Inferences in KL-ONE-based Knowledge Representation Systems
21 pages

RR-90-07
Elisabeth André, Thomas Rist: Wissensbasierte Informationspräsentation:
Zwei Beiträge zum Fachgespräch Graphik und KI:
1. Ein planbasierter Ansatz zur Synthese illustrierter Dokumente
2. Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen

24 Seiten

RR-90-08
Andreas Dengel: A Step Towards Understanding Paper Documents
25 pages

RR-90-09
Susanne Biundo: Plan Generation Using a Method of Deductive Program Synthesis
17 pages

RR-90-10
Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann: Concept Logics
26 pages

RR-90-11
Elisabeth André, Thomas Rist: Towards a Plan-Based Synthesis of Illustrated Documents
14 pages

RR-90-12
Harold Boley: Declarative Operations on Nets
43 pages

RR-90-13
Franz Baader: Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles
40 pages

RR-90-14
Franz Schmalhofer, Otto Kühn, Gabriele Schmidt: Integrated Knowledge Acquisition from Text, Previously Solved Cases, and Expert Memories
20 pages

RR-90-15
Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16
Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17
Stephan Busemann: Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01
Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02
Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03
B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04
Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05
Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06
Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents A Plan-Based Approach
11 pages

RR-91-07
Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters: The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka: Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos: The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39pages

RR-91-31
H.-U. Krieger , J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

DFKI Technical Memos

TM-90-02
Jay C. Weber: The Myth of Domain-Independent Persistence
18 pages

TM-90-03
Franz Baader, Bernhard Hollunder: KRIS: Knowledge Representation and Inference System -System Description-
15 pages

TM-90-04
Franz Baader, Hans-Jürgen Bürckert, Jochen Heinsohn, Bernhard Hollunder, Jürgen Müller, Bernhard Nebel, Werner Nutt, Hans-Jürgen Profitlich: Terminological Knowledge Representation: A Proposal for a Terminological Logic
7 pages

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS:
The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley:
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel , Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

DFKI Documents

D-90-03
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Abschlußbericht des Arbeitspaketes PROD
36 Seiten

D-90-04
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch
69 Seiten

D-90-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Formalismus zur Repräsentation von Geo-metrie- und Technologieinformationen als Teil eines Wissensbasierten Produktmodells
66 Seiten

D-90-06
Andreas Becker: The Window Tool Kit
66 Seiten

D-91-01
Werner Stein , Michael Sintek: Relfun/X - An Experimental Prolog Implementation of Relfun
48 pages

D-91-03
Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 pages

D-91-09
David Powers and Lary Reeker (Eds):
Proceedings MLNLO´91 - Machine Learning of Natural Language and Ontology
211 pages

Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW´91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages

Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
*Thilo C. Horstmann:*Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
H^{iera}C$_{on}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: KRIS : Knowledge Representation and Inference System
- Benutzerhandbuch -
28 Seiten

A Diagnostic Tool for German Syntax

John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann and Judith Klein

RR-91-18

Research Report