# DiTo - Ein Diagnostik-Werkzeug für die syntaktische Analyse *

**Judith Klein[t], Ludwig Dickmann[#], Abdel Kader Diagne, John Nerbonne und Klaus Netter[t]**

[t]Deutsches Forschungszentrum für Künstliche Intelligenz, GmbH
Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, FRG Telephone: (+49 681) 302-5303, e-mail: klein@dfki.uni-sb.de

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

Abstrakt

In dieser Arbeit wird ein Testwerkzeug für die Fehlerdiagnose bei Syntaxkomponenten natürlichsprachlicher Systeme vorgestellt. Wir diskutieren kurz die Relevanz von Testwerkzeugen für natürlichsprachliche Systeme und befürworten die Idee modularer Testtools. In diesem Rahmen stellen wir einen Ansatz vor, der im Bereich der Syntaxkomponente angesiedelt ist. Mit unserem Diagnostik-Tool unternehmen wir den Versuch, einen Datenkatalog zu erstellen, der die wesentlichen Phänomene deutscher Syntax erfaßt, um die Fehlerdiagnose zu unterstützen. Bisher beinhaltet der Datenkatalog die Bereiche *Verbrektion, Koordination* und - noch nicht ganz vollständig - *Funktionsverbgefüge.* Wir arbeiten mit anderen Gruppen[1] zusammen, die weitere Syntaxthemen entsprechend den Richtlinien unseres Ansatzes erarbeiten. Damit ausgewählte Syntaxgebiete separat abgetestet werden können, sind die Daten in einer relationalen Datenbank organisiert.

**Abstract**

In this paper we present a testing tool for the diagnosis of errors in NLP Systems. We discuss briefly the relevance of testing tools for NLP Systems and advocate the idea of modular testing tools. liere we present an approach for the syntax component of NLP Systems. Our diagnostic tool for German syntax is an effort to construct a catalogue of syntactic data exemplifying the major syntactic patterns of German that Supports the diagnosis of errors. Up to now, the catalogue contains the areas *verbal government, coordination* and - although not yet completed - *fixed verbal structures. We* cooperate with other groups[2] that work on further syntactic phenomena according to the ideas of DiTo. To allow systematic testing of specific areas of syntax the data are organised into a relational database.

## Motivation und Ziele

Bei der Entwicklung natürlichsprachlicher Systeme müssen den Entwicklern Testmengen zur Verfügung stehen, anhand derer sie die Performanz der Systeme kontrollieren können. Durch die modulare Architektur natürlichsprachlicher Systeme besteht die Möglichkeit,

*Diese Arbeit wurde durch einen Forschungszuschuß, ITW 9002 0, vom Deutschen Bundesministerium für Forschung und Technologie an das DISCO Projekt am DFKI und durch IBM Deutschland Projekt LILOG-SB an der Universität des Saarlandes finanziell unterstützt.

[1]IAI (Institut für angewandte Informationswissenschaft), Projekt EUROTRA in Saarbrücken und Institut für Computerlinguistik an der Universität Koblenz

[2]IAI (Institute for Applied Information Science), project EUROTRA in Saarbrücken and Institute for Computational Linguistics at the University of Koblenz

einzelne Komponenten und Teilkomponenten separat zu testen und entsprechend zu modifizieren. Leider existieren oftmals entweder nur unzureichende Testmengen oder ausführlich erarbeitete Testdaten sind nur für ein spezielles System angelegt worden, so daß sie für ein zweites System nicht brauchbar sind. Aus dieser Überlegung heraus entstand die Idee, eine Testmenge zu entwickeln, die im Bereich der *Syntaxkomponente* als Kontrollmenge eingesetzt werden kann. Unser Diagnostik-Tool DiTo ist nun der Versuch, einen linguistischen Datenkatalog zu erstellen, mit dem Ziel, möglichst alle wesentlichen Bereiche der deutschen Syntax anhand von Beispieldaten *systematisch* abzudecken. Der Katalog hat folgende Aufgaben:

- Debugging : Fehler innerhalb der syntaktischen Verarbeitung können leichter loka lisiert und benannt werden, wenn eine empirische Grundlage für die Fehlerdiagnose zur Verfügung steht.
- Konsistenzhaltung : Anhand der Daten kann überprüft werden, ob sich die Ab deckung eines syntaktischen Phänomens verändert hat, nachdem z.B. die Gramma tik an anderer Stelle modifiziert wurde.
- Monitoring der Systemperformanz : Die Bearbeitung einzelner Syntaxbereiche kann durch die regelmäßige Anwendung der Testsätze gezielt kontrolliert werden.

Von anderen Arbeiten in diesem Bereich unterscheidet sich unser Ansatz in zwei Punkten:

- Die Beispieldaten sind systematisch erstellt und nicht aus Texten extrahiert, um eine möglichst genaue Kontrolle über die Testdaten zu haben
- Die Beispieldaten sind mit syntaktischen Annotationen versehen, die das jeweilige Syntaxphänomen weitgehend beschreiben und mit allgemein-syntaktischen Informa tionen, die die Oberflächenstruktur der Sätze betreffen.

Bei der Wahl der Annotierungen sind wir von folgenden Überlegungen ausgegangen: Der Datenkatalog wird nützlicher sein, wenn Beispielsätze zu einem ausgewählten Bereich herausgezogen werden können, d.h. die syntaktischen Merkmale, die mit einem Beispielsatz verknüpft sind, betreffen *ein* spezielles syntaktisches Phänomen. Zusätzlich dienen die allgemein-syntaktischen Informationen dazu, die Systeme anhand sehr genauer und leicht kontrollierbarer Daten auf Genauigkeit innerhalb der syntaktischen Analyse hin zu überprüfen.

## Die Datengrundlage

Unsere Datensammlung besteht aus systematisch gebildeten Beispielsätzen, die mit syntaktischen Merkmalen annotiert sind. Um bei dem angestrebten Einsatz der Testmenge zu erreichen, daß alle und nur die richtigen Sätze verarbeitet werden und sinnvolle Toleranzgrade bei der Fehlerbehandlung herausgearbeitet werden können, umfaßt die Datensammlung auch eine Zusammenstellung ungrammatischer Sätze, die ebenfalls systematisch für die jeweiligen Syntaxbereiche erarbeitet wurden. Die DiTo-Datenstruktur soll an den Beispielen *Verbrektion* und *Koordination* verdeutlicht werden. Generell haben wir versucht, so viele Parameter wie möglich bei der Konstruktion der Beispielsätze einzuschränken, um unnötige Mehrdeutigkeiten zu vermeiden. Zum Beispiel beschränken wir uns bei den NPs soweit wie möglich darauf, den bestimmten Artikel zu verwenden und morphologische Ambiguitäten zwischen Feminin in Nominativ und Akkusativ und Neutrum in Nominativ und Akkusativ dadurch auszuschließen, daß wir diese Formen nicht verwenden.[3]

Bei der *Verbrektion* ging es darum, alle Kombinationen obligatorischer Komplemente zusammenzustellen, die die deutschen Satzschemata bilden. Zu allen möglichen Kombinationen nominaler, präpositionaler, adjektivischer und sententialer Komplemente haben

[3]Wird das Thema *NP-Syntax* bearbeitet, gelten diese Bestimmungen natürlich nicht. Bei einigen Beispielsätzen konnten nicht alle Kriterien exakt eingehalten werden; dies ist dann in einem besonderen Kommentarfeld vermerkt.

wir Beispielsätze konstruiert. Das Ergebnis ist eine Liste von ca. 70 Kombinationen, die durch etwa 220 Sätze (440 mit den ungrammatischen Beispielen) illustriert sind. Einige Beispieldaten:

- Kombinationen nominaler Komplemente:
  "Der Manager gibt dem Studenten den Computer."
- Nominale Komplemente kombiniert mit Sententialen Komplementen:
  "Der Vorschlag dient dazu, den Plan zu erklären."

Jede Komplement-Kombination ist durch mindestens einen Beispielsatz veranschaulicht. Außerdem gehört zu jedem Rektionstyp eine Zusammenstellung ungrammatischer Sätze. Die zweite Testmenge beschreibt Phänomene der *Satzkoordination,* wie zum Beispiel "Gapping", "Rechts-" und "Linkstilgung". Die beiden Ausgangssätze, die dieser Datensammlung zugrunde liegen, sind: *Ich glaube, daß der Professor der Sekretärin den Blumenstrauß schenkte, und der Student dem Kommilitonen den Roman verkaufte.* als Beipiel für Verb-Letzt-Stellung und *Der Professor schenkte der Sekretärin den Blumenstrauß , und der Student verkaufte dem Kommilitonen den Roman,* als Beispiel für Verb-Zweit-Stellung. In diesen beiden Sätzen wurden systematisch alle möglichen Tilgungsvariationen vorgenommen und - um einer kombinatorischen Explosion vorzubeugen - alle möglichen Permutationen nur im *zweiten* Konjunkt durchgeführt. Das Ergebnis ist eine Liste von ca. 100 grammatischen und 430 ungrammatischen Beispielsätzen. Einige Beispieldaten:

- Gapping: (Tilgung des Hauptverbes und einer NP im zweiten Konjunkt)

"Der Professor schenkte der Sekretärin den Blumenstrauß und der Student den Roman."

- Linkstilgung:

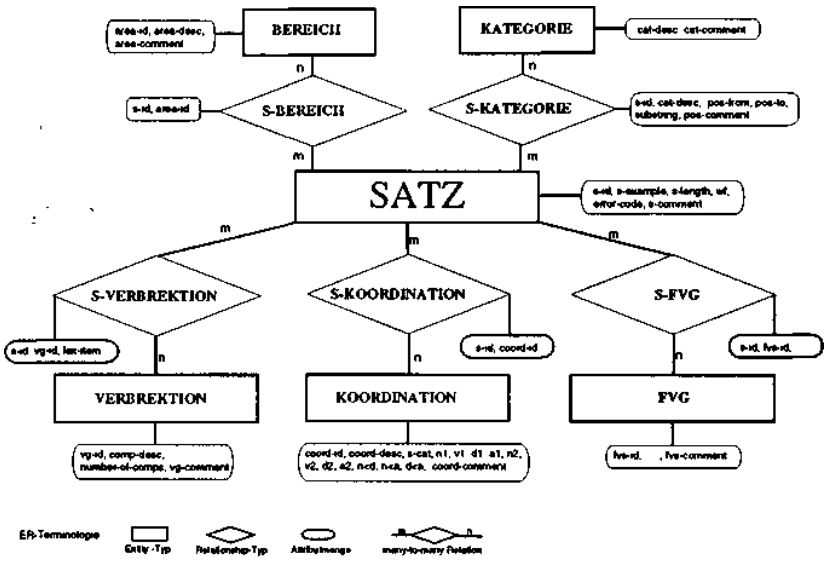
"Ich glaube, daß der Professor der Sekretärin den Blumenstrauß schenkte und den Roman verkaufte."

### Die Organisation der Daten in der relationalen Datenbank

Die Daten sind so in der Datenbank organisiert, daß entweder Beispielsätze anhand vorgegebener syntaktischer Merkmale erfragt werden können, oder daß die syntaktische Beschreibung konkreter Sätze ausgegeben werden kann. Mehrere Gründe sprechen dafür, das linguistische Material in einer relationalen Datenbank zu organisieren:

- Die Testmenge ist klar strukturiert und erleichtert so den Zugriff auf die Daten.
- Auf diese Weise ist es leicht, die Einträge konsistent zu halten.
- Den interessierten Gruppen stehen verschiedene logische Sichten auf die Daten zur Verfügung.
- Wir haben so gute Voraussetzungen, um die Datenbank um neue Syntaxbereiche zu erweitern.

Unsere Datenbank wurde in awk programmiert, weil awk als *public domain*-Software verfügbar ist und sowohl unter UNIX als auch unter MS-DOS läuft, was unserer Idee, dieses Test-Tool letztlich an alle interessierten Gruppen weiterzugeben, entgegenkommt. Außerdem erlaubt awk eine Stringbehandlung mit besonderen Matchingfunktionen, die Standard-Datenbanksoftware nicht bereitstellt, und die für unser Datenmaterial sehr wünschenswert ist. Andererseits bietet awk keine *schnellen* Zugriffsmethoden auf die gespeicherten Informationen, was für unsere Zwecke aber auch nicht im Vordergrund steht. Zusätzliche Tools wie z.B. eine Anfragesprache AQL, die mit Hilfe der Werkzeuge LEX, einem lexikalischen Analyse-Generator, und YACC, einem Parser-Generator, entwickelt wurde, konnten leicht in die Systemumgebung eingebunden werden. Das konzeptuelle Schema der Datenbank wird in Abbildung l illustriert:




Die zentrale Relation in der Datenbank ist SATZ. Sie umfaßt Informationen über die eindeutig zugewiesene Satznummer, das Satzbeispiel, die Salzlauge, die Grammatikalität des Satzes, den möglichen Syntaxfehler und mögliche Besonderheiten des Satzes, die dann im Kommentarfeld eingetragen werden. Die Einträge in SATZ haben folgende Form:

| s-id | s-example | s-length | wf | error-code | s-comment |
|---|---|---|---|---|---|
| 1168 | der manager beurteilt den Studenten danach, ob der vortrag gut ist | 11 | 1 | 0 | null |
| 3021 | *der professor schenkte der sckretaerin einen blumenstrauss und der Student verkaufte dem kommilitonen | 13 | 0 | -10 | rechts tilgung |

In den Relationen VERBREKTION, KOORDINATION und FVG (Funktionsverbgefüge) stehen die Attribute, die für diese syntaktischen Phänomene entscheidend sind. Durch S-VERBREKTION, S-KOORDINATION und S-FVG werden die Satzbeispiele mit den syntaktischen Merkmalen verknüpft. S-BEREICH ordnet die Satzbeispiele den in BEREICH eingetragenen Syntaxbercichen zu. In S-KATEGORIE sind die allgemein-syntaktischen Informationen zu den Satzbeispielen gespeichert.

Folgende Anfragen zeigen die Verbindungen der Daten untereinander. In den ersten beiden Anfragen werden Beispielsätze erfragt, die bestimmte syntaktische Merkmale haben.

- Zeige alle Sätze, die eine Nominativ-NP, eine Dativ-NP und eine Akkusativ-NP haben.

retrieve s-id s-example where comp-desc = "nom dat acc.
1022 der manager gibt dem Studenten den Computer.
1023 der manager verdankt dem praesidenten den Computer.
1235 *der manager gibt.
1236 *der manager gibt dem Studenten.

- Zeige die grammatischen Sätze, in denen im zweiten Konjunkt Gapping auftritt.[4]

[4]Der Wert 2 in der Anfrage zeigt *Tilgung* an.

```
retrieve s-id s-example where v2 = 2 and (n2 = 2 or d2 = 2 or a2 =2)
and wf = l
3025 der professor schenkte der Sekretärin den blumenstrauss und dem
kommilitonen den roman.
3026 der professor schenkte der Sekretärin den blumenstrauss und den
roman dem kommilitonen.
```

*Das* letzte Beispiel zeigt eine Anfrage, die von einem konkreten Satz ausgehend allgemein-syntaktische Merkmale dieses Satzes erfragt.

- Zeige die Position der NP's in den Sätzen mit der Komplement-Beschreibung "nom_dat_acc".

```
retrieve s-id cat-desc cat-position substring where cat-desc = "np" and
comp-desc =  "nom_dat_acc "
1022 np l 2 der manager 1022 np 4 5 dem
Studenten 1022 np 6 7 den Computer 1022
finite_matrix_verb 3 3 gibt
```

## Ausblick

Unser Diagnostik-Tool ist mit einer flexiblen Schnittstelle versehen, die es ermöglicht, die Daten den Systemen, die die Testsätze einsetzen wollen, zugänglich zu machen. Um das endgültige Ziel, alle Bereiche deutscher Syntax systematisch abzudecken, erreichen zu können, brauchen wir die Unterstützung anderer Forschungsgruppen, die weitere Bereiche von syntaktischen Phänomenen erarbeiten. Kontakte und Zusammenarbeit bestehen bereits mit dem IAI in Saarbrücken[5] und dem Institut für Computerlinguistik in Koblenz[6].

## Literatur


[1] Abdel Kader Diagne: DiTo - *DMS. The DiTo Database Management System. Concepts, Implementation Issues and User Guide.* To appear as: DFKI Technical Document, DFKI, Saarbrücken, 1992.

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

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

[4] Judith Klein and Ludwig Dickmann: DiTo - *Datenbank. Daten-Dokumentation zu Verbrektion und Koordination.* To appear as: DFKI Technical Document, DFKI, Saarbrücken 1992.

[5] Martha Palmer and Tim Finin: Workshop on the Evaluation of Natural Language Processing Systems. In: *Computational Linguistics 16(3),* 1990, pp.175-181.

[6] 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.

[7] Martin Volk and Hanno Ridder: GTU - eine Grammatik Testumgebung mit Testsatzarchiv. To appear in: *LDV-Forum 1.1992*


[5] Brigitte Krenn arbeitet am IAI in Saarbrücken an Funktionsverbgefügen.

[6] Martin Volk vom Institut für Computerlinguistik in Koblenz hat vorgeschlagen, Relativsatze zu erarbeiten