Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document
D-93-07

Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse

Klaus-Peter Gores, Rainer Bleisinger

Mai 1993

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

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

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
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 and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, 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
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

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.

Friedrich J. Wendl
Director

Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse

Klaus-Peter Gores, Rainer Bleisinger

DFKI-D-93-07

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9003 0).

Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse

Klaus-Peter Gores & Rainer Bleisinger

Deutsches Forschungszentrum für Künstliche Intelligenz

Postfach 2080, 6750 Kaiserslautern

Tel.: (+49 631) 205 3481

email: bleising@dfki.uni-kl.de

Zusammenfassung

In dieser Papier wird die koordinierende Komponente eines Systems zur erwartungsgesteuerten Textanalyse auf der eingeschränkten Domäne deutscher Geschäftsbriefdokumente vorgestellt. Dazu wurden wesentliche Konzepte und Datenstrukturen zur Modellierung der Domäne, das Nachrichtenmodell, entwickelt (siehe [Gores & Bleisinger 92]). Mit diesem Nachrichtenmodell steuert die Komponente die Textextraktion der Informationen eines vorliegenden Briefdokumentes. Sie wird in ihrer Arbeit von Spezialisten, sogenannten Substantiierern, unterstützt, die auf dem Text arbeiten. Dazu muß intensiver Nutzen von den Informationen eines Lexikons gemacht werden. Die Repräsentation des Ergebnisses erfolgt in einer Form, die eine weitere Verarbeitung, wie die semantische Interpretation und eine darauf aufbauende Generierung neuer Aktionen begünstigt.

Inhaltsverzeichnis

1 Einleitung

Heutzutage ist der Computer ein fester Bestandteil in einem modernen Büro. Die Generierung und Verarbeitung von Text mittels Computer ist bereits eine Selbstverständlichkeit. Doch trotz der enormen Fortschritte beim elektronischen Medium bleibt die Abhängigkeit vom Papier weiterhin bestehen. Diese Tatsachen begründen die Notwendigkeit von intelligenten Hilfsmitteln, die die Kluft zwischen Papier und Computer schließen.

In dem am DFKI durchgeführten Projekt ALV (Automatisches Lesen und Verstehen) wird ein System zur Analyse von einseitigen Papierdokumenten, speziell von deutschsprachigen Geschäftsbriefen entwickelt. Hauptanliegen ist dabei die Transformation von gedruckter Information in eine symbolische Repräsentation, die mittels Computer handhabbar und weiter verarbeitbar ist. Insbesondere soll das System in der Lage sein, Text so zu lesen, daß seine Bedeutung in gewissem Umfang automatisch extrahiert werden kann. Analog zum menschlichen Lesen wird eine enge Verflechtung zwischen automatischem Erkennen und automatischem Verstehen textueller Information in Papierdokumenten verfolgt. Der Nutzen eines solchen Systems soll auf lange Sicht ein verringerter Aufwand zur Bewältigung der eingehenden Geschäftspost sein, etwa indem die Verteilung oder die Ablage unterstützt wird.

Ein Überblick über den prinzipiellen Ablauf der in ALV durchgeführten wissensbasierten Dokumentanalyse ist in Abbildung 1 kurz skizziert. Dabei sind drei wichtige Phasen zu identifizieren: die Strukturanalyse, die Texterkennung und die partielle Textanalyse.

Umfassende Beschreibungen, vor allem von Strukturanalyse und Texterkennung, finden sich in [Dengel et al 92a] und [Dengel et al 92b]. Nachfolgend wird nur ein kurzer Überblick gegeben und auf vertiefende Literaturstellen verwiesen.

In ALV beginnt die Dokumentanalyse mit der elektronischen Erfassung des eingehenden Papierdokumentes durch einen *Scanner*. Das Ergebnis ist eine *Bit-Map*, d.h. eine zweidimensionale Matrix mit binären Werten (*Pixel*). Diese repräsentiert die graphische Information des Dokumentes als schwarz-weiß-Bild. In einem Vorverarbeitungsschritt wird der Drehwinkel des gescannten Dokumentes bestimmt ([Dengel & Schweizer 89]) und bei Bedarf (Drehwinkel ungleich 0) wird eine zur weiteren Verarbeitung notwendige Korrektur dieses Winkels vorgenommen ([Ali 92]).

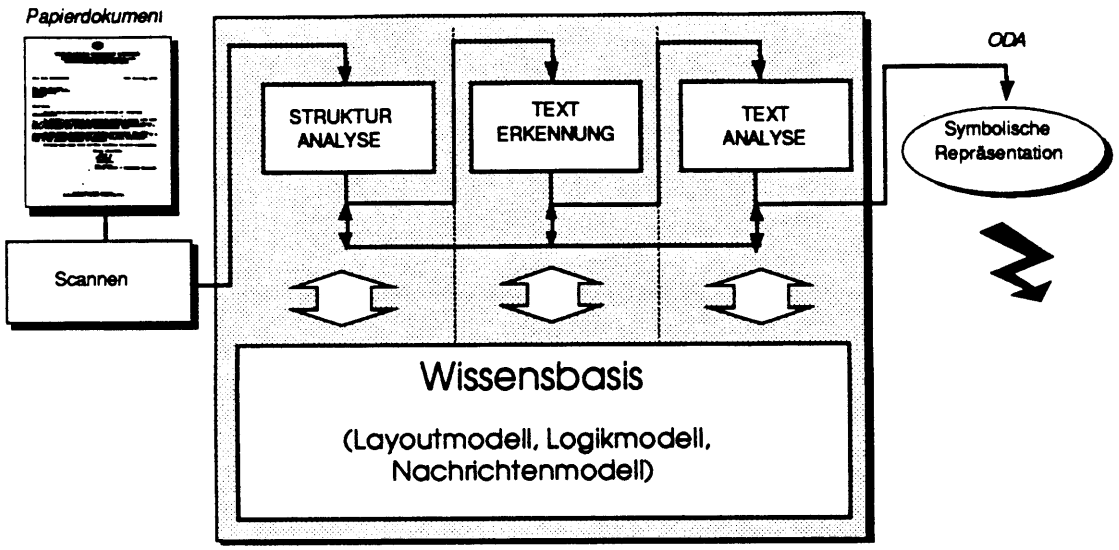


Abbildung 1: Ablauf der Dokumentanalyse in ALV

Diese unstrukturierte Datenmenge wird in der folgenden Phase der Strukturanalyse weiterverarbeitet. Zuerst werden in der Matrix sogenannte *Zusammenhangsgebiete*, d.h. benachbarte Schwarzpixelbereiche, berechnet. Daraus werden in der *Segmentierung* die hierarchisch aufgebauten *Layoutobjekte* gebildet, z.B. Zeichen, Worte, Zeilen oder Blöcke ([Fein et al 92]). Durch ein *logical labeling* werden den so ermittelten Layoutobjekten unter Einbeziehen von geometrischem Wissen sogenannte *logische Objekte*, z.B. Briefrumpf, Absender oder Betreff, zugeordnet ([Dengel 92]).

Auf diesem Ergebnis setzt eine erwartungsgesteuerte Texterkennung auf, die für alle Layoutobjekte vom Typ Zeichen deren codierte Darstellung berechnet, z.B. als ASCII-Wert. Die Erkennung geschieht aufgrund visueller Merkmale. Doch die Zeichen sollen nicht nur isoliert betrachtet werden, sondern die redundante Information des Wortaufbaus soll ebenfalls genutzt werden ([Boon 92], [Molter 92]). Die Verwendung eines strukturierten Lexikons, dessen spezifische Struktur auch auf die logischen Objekte zugeschnitten ist, soll dabei die Erkennung des Textes innerhalb der logischen Objekte auf der Wortebene unterstützen ([Dengel et al 92c], [Hoch & Malburg 92]).

Bei der Textanalyse soll in Verbindung mit logischen Objekten eine partielle Analyse des so erkannten Textes durchgeführt werden. Als Besonderheit ist dabei zu beachten, daß der Text nicht vollkommen korrekt erkannt wird. So entstehen für ein Wort oftmals mehrere Alternativen, oder aber es kann gar kein Vorschlag gemacht werden und es entstehen Lücken. Diese "Fehler" in der Texterkennung sind somit von allen nachfolgenden Analyseverfahren zu berücksichtigen.

Im Bereich der Textanalyse reicht das Spektrum von Ansätzen mit formalen Sprachen bis hin zu linguistisch orientierten Ansätzen (z.B. Transformations-Grammatiken, [Chomsky 56]). Andere Wege zur Sprachanalyse beziehen das Wissen über den Kontext der Sprache mit ein (Kasus-Grammatiken, [Fillmore 71], Diskurs-Repräsentations-Grammatiken [Kamp 88]).

Es hat sich aber leider gezeigt, daß alleine mit einer formalen Grammatik noch nicht einmal die Syntaxanalyse völlig gelingt, ein Verstehen von Sprache liegt damit in weiter Ferne. Dies liegt zum Teil an der starken Abhängigkeit der Analyse von der syntaktischen Ausprägung der Eingabe: Überlicherweise beginnt die Analyse auf der syntaktischen Ebene (*parsen*) und geht anschließend zur semantischen Analyse über, wenn kein Fehler aufgetreten ist. Die semantische Analyse befindet sich damit in völliger Abhängigkeit von der syntaktischen. Viele Analysevorgänge scheitern allein aufgrund einer dem System unverständlichen Satzstruktur; der Satz kann nicht verstanden werden. Wenn die semantische Analyse mit der syntaktischen überlappend ausgeführt wird, können auch syntaktisch zweifelhafte Konstrukte interpretiert werden. Um ein robusteres Verhalten zu erreichen, sollte die semantische Analyse möglichst unabhängig vom Ergebnis der Syntaxanalyse sein. Die Syntax darf dann in gewissen Grenzen unvollständig oder fehlerhaft sein. Die Forderung nach der Unabhängigkeit läßt sich noch schärfer formulieren: die syntaktische Analyse soll von der semantischen Analyse kontrolliert werden. Dies ist die Idee der erwartungsgesteuerten Textanalyse, die durch die Angabe einer starken Erwartungshaltung über den Inhalt eines Textes realisiert wird ([DeJong 79,82], [Bobrow 77], [Lebowitz 83,85).

Die Vorteile eines erwartungsgesteuerten Verfahrens für die Textanalyse liegen in der Effizienz und vor allem in der geringen Anfälligkeit gegenüber fehlerhaften und unvollständigen Eingaben. Die Eingabe der Textanalyse in ALV weist als Ergebnis der Texterkennung gerade solche Lücken und Alternativen auf, die es besonders zu beachten gilt.

2 Das Modell einer erwartungsgesteuerten Textanalyse

In diesem Kapitel werden der Ablauf der erwartungsgesteuerten Textanalyse, wie sie in ALV realisiert ist, und die beteiligten Komponenten im Überblick beschrieben. Das Modell der erwartungsgesteuerten Textanalyse kann grob in fünf Komponenten gegliedert werden (siehe Abbildung 2): der Predictor und die Substantiierer als aktive Einheiten, das Nachrichtenmodell, das Lexikon und der aktuelle Kontext eher als passive Datenstrukturen.

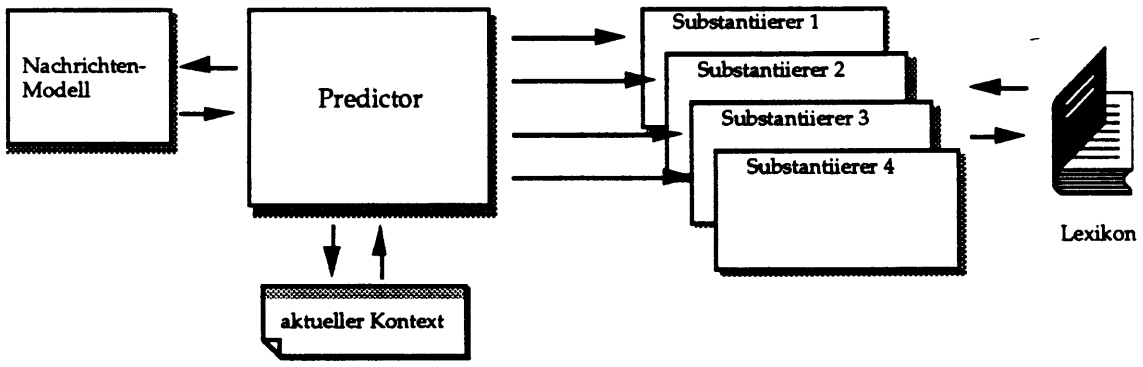


Abbildung 2: Die Komponenten des Systems

Die Analyse eines Briefdokumentes mit diesem System hat den folgenden Ablauf: Um aus dem vorliegenden Briefdokument den Inhalt zu extrahieren, generiert der Predictor Erwartungen über den Text. Dazu benutzt er das Nachrichtenmodell, in dem das Wissen über Briefdokumente repräsentiert ist. Mit dieser Erwartung beauftragt der Predictor einen oder mehrere Substantiierer, das vorliegende Briefdokument mit Hilfe von einfacheren inhaltlichen Analysen zu untersuchen. Die Eingabetexte werden dabei nicht vollständig analysiert, sondern gezielt überflogen. Nur die Bereiche werden betrachtet, in denen laut den Erwartungen des Predictors wichtige Inhalte erwartet werden; alle übrigen Textinformationen werden ignoriert. Die Substantiierer werden durch syntaktische und semantische Informationen des Lexikons unterstützt. Das Ergebnis der Substantiierer wird an den Predictor übergeben, durch ihn interpretiert und in den aktuellen Kontext integriert. Mit dem so erweiterten Wissen generiert der Predictor neue Erwartungen und das Verfahren beginnt mit einem neuen Aufruf der Substantiierer. Durch die wiederholte Ausführung dieser

Schleife wird aus dem Briefdokument schrittweise dessen textueller Inhalt als interne Repräsentation aufgebaut. Damit ist der Ablauf einer erwartungsgesteuerten Textanalyse vom Zusammenspiel der vorhersagenden, koordinierenden Komponente eines Predictors und der subordinierten Substantiierer geprägt.

Dadurch, daß sehr viel über den erwarteten Inhalt eines Dokumentes bekannt ist, kann die Analyse robuster verlaufen. Je umfangreicher das Wissen im Nachrichtenmodell ist, das über die möglichen Inhalte eines Textes zur Verfügung steht, um so gezielter kann eine Erwartung ausgesprochen werden. Das Ziel einer erwartungsgesteuerten Textanalyse besteht also nicht darin, jeden möglichen Eingabetext zu erkennen. Stattdessen muß die Eingabe auf die Domäne beschränkt sein, für die die Erwartungen angegeben wurden. Nur Eingaben, die zu den Erwartungen passen, können auch analysiert werden.

Die an der erwartungsgesteuerten Textanalyse beteiligten Komponenten werden in den folgenden Abschnitten und Kapiteln kurz vorgestellt und wichtige Begriffe eingeführt. Begonnen wird mit dem Lexikon, woran sich eine kurze Diskussion über das Nachrichtenmodell anschließt. Das Nachrichtenmodell wird ausführlich in [Gores & Bleisinger 92] erklärt. Die Aufgaben der Substantiierer werden zusammen mit möglichen Verfahren anschließend erörtert. Der Predictor und der aktuelle Kontext werden ebenfalls nur

skizziert, die ausführliche Beschreibung dieser Komponenten findet sich in Kapitel 3. Abrundend wird in diesem Kapitel der prinzipielle Ablauf der erwartungsgesteuerten Analyse mit diesen Bestandteilen detaillierter vorgestellt.

2.1 Das Lexikon

Die während der Dokumentanalyse zur Verfügung stehenden Worte sind in einem Lexikon zusammengefaßt. Für jeden Worteintrag sind die verschiedensten Informationen gespeichert, je nach den Bedürfnissen der unterschiedlichen Analysephasen und Analyseaufgaben. Um ein effektiveres Benutzen eines riesigen Wortbestandes zu ermöglichen, bietet sich eine Strukturierung des Lexikons an.

Dazu werden Worte, denen bestimmte Eigenschaften gemein sind, zu Wortmengen zusammengefaßt. Damit bietet das Lexikon verschiedene Sichten, sogenannte *lexical views*, auf spezielle Wortmengen als Teile des gesamten Bestandes an. Beispiele von lexical views sind: syntaktische Sichten[1], z.B. `LV-verb`, `LV-noun`, etc.; semantische Sichten, die Ortsnamen, Lebewesen etc. umfassen; domänen-spezifische Sichten wie `LV-product` oder `LV-customer`.

Aus den Bedürfnissen der erwartungsgesteuerten Textanalyse ergeben sich für das Lexikon Anforderungen nach drei verschiedene Ausprägungen der lexical views:

- syntaktische Lexikonsichten: auch die erwartungsgesteuerte Analyse kommt nicht

Komparationen), die anderweitig nur schwer oder gar nicht gewonnen werden kann. Eine Disambiguierung semantischer Mehrdeutigkeiten wird vereinfacht.

- (schwache) semantische Lexikonsichten: den Bedingungen der Conceptual Dependency ([Schank 72, 73]) genügende Sichten wie Aktionen, Agenten, Objekte etc..
- Lexikonsichten der Domäne: höher strukturierte semantische Sichtweisen. Darin können hierarchische Beziehungen der Vererbung (`is-a`), Beziehungen zwischen Konzepten (`has-parts, is-part-of`) und anderes mehr ausgedrückt werden. Außerdem sind im Lexikon Sichten zu definieren, die spezielle Begriffe der Geschäftsbriefe einordnen. Typische Beispiele dafür sind lexical views für Kunden, Lieferanten, Produkte oder Ortsnamen.

Die Anforderungen an die Schnittstelle des Lexikons zu den Substantiierern wird von den Bedürfnissen des Predictors motiviert, der möglichst viel über die Worte erfahren will, die die Erwartungen erfüllt haben. Die Antwort, die ein Substantiierer vom Lexikon erhält, muß daher

- den Verweis auf den Lexikoneintrag enthalten. Damit kann sich der Predictor alle nötigen Informationen besorgen.
- die Semantik angeben, die diesem Wort zugeordnet ist, also die lexical views, zu denen das Wort gehört.
- die syntaktischen und morphologischen Begründungen liefern, die von der Stammform zur Textfassung geführt haben, also vorliegende Deklinations-, Konjugations- oder Komparationsklassen[2].

Mit diesem Anforderungskatalog, der die verschiedenen Arten der lexical views umreißt, sind die wesentlichen Elemente beschrieben, die das Lexikon aus der Sicht des Predictors enthalten sollte: Die syntaktische Information, die in den Nachrichtentypen und von den Substantiierern benötigt wird; die semantische Information, aufgeteilt in allgemeines und Domänenwissen, die in den Nachrichtentypen und vom Predictor benutzt wird.

2.2 Das Nachrichtenmodell

Das gesamte Wissen des Analysesystems über Geschäftsbriefdokumente wird in der Datenstruktur des *Nachrichtenmodells* gespeichert. Mit diesem Wissen kann der Predictor gezielt Erwartungen über die Inhalte von Geschäftsbriefen aufstellen. Das Nachrichtenmodell enthält die aufeinander aufbauenden Definitionen von *Aspekten, Slots, Nachrichtenelementen, Nachrichtentypen* und *Multi-Nachrichtentypen*. Diese Definitionen dienen zur Repräsentation der Erwartungen verschiedener Bestandteile von Geschäftsbriefen, unterschiedlicher Klassen von Geschäftsbriefen und ganzer Korrespondenzketten von Geschäftsbriefen.

---

2 Im Projekt ALV wird zur Zeit ein Vollformenlexikon verwendet.

Wenn man Geschäftsbriefe untersucht, stellt man leicht fest, daß sie sich in verschiedene Klassen einteilen lassen. Typische Einteilungen umfassen z.B. die Klassen Anfrage, Angebot, Danksagung, Bestellung, Lieferung und Rechnung. Diese Klassen werden im Nachrichtenmodell durch die Nachrichtentypen dargestellt. Jeder Klasse eines Briefdokumentes entspricht also ein bestimmter Nachrichtentyp.

Ein Geschäftsbrief stellt aber keinen homogenen Text dar, sondern kann in einzelne Bestandteile zerlegt werden. Die Bestandteile einer Bestellung umfassen die Adresse, mehrere Bestellungen und ein Lieferdatum. Für jeden dieser Bestandteile gibt es in dem Nachrichtentyp ein Nachrichtenelement, das dem Bestandteil des Briefes entspricht. Nachrichtenelemente werden in einer erweiterten Conceptual Dependency-Notation beschrieben ([Gores & Bleisinger 92]). Auch diese Bestandteile des Briefes können noch weiter differenziert werden: eine Adresse besteht aus Name, Straße, Postleitzahl und einem Ort. Diesen Basiselementen von Briefen, die meistens nur aus einem Wort bestehen, entsprechen die einzelnen Slots eines Nachrichtenelementes.

Diesen Slots sind Constraints zugeordnet, die festlegen, welche Art von Wort eingetragen werden darf. Die Constraints beziehen sich stets auf lexical views, die im Lexikon definiert werden können. Zusätzlich zu den inhaltlichen Constraints werden auch Anforderungen an das Layout oder die Zuordnung zu einem logischen Objekt gestellt. Damit wird die Erwartungshaltung der einzelnen Slots und damit des gesamten Nachrichtentyps aufgestellt.

Innerhalb eines Geschäftsbriefs oder einer Korrespondenzkette tauchen oft mehrere Dokumentklassen in einer festen Reihenfolge auf. Ein typischer Ablauf beginnt mit einem Angebot, einer Bestellung, der anschließenden Lieferung und Rechnung. Dieser Zusammenhang mehrerer Klassen von Briefen wird durch die Zusammenfassung der entsprechenden Nachrichtentypen in einem Multi-Nachrichtentyp realisiert.

Die Definitionen der Nachrichtentypen durch ihre Nachrichtenelemente und Slots geben an, wie die Briefdokumente strukturiert sind. Damit stellen sie die Erwartungen bereit, die der Predictor auf die Eingabedokumente der Analyse anwenden kann. Eine Erwartung eines Nachrichtenelementes gilt als bestätigt, wenn die Slots gefüllt werden konnten, ein erwarteter Nachrichtentyp wenn seine Nachrichtenelemente bestätigt wurden.

2.3 Die Substantiierer

Die Aufgabe der Substantiierer besteht darin, den Erwartungen des Predictors Leben einzuhauchen. Die Erwartungen, die der Predictor aus den Nachrichtentypen generiert, müssen an geeignete Worte oder Phrasen des Eingabedokumentes gebunden werden. Eine befriedigte Erwartungshaltung besteht also in einer an Worte des Textes gebundenen Erwartung. Um diese Aufgabe zu lösen, analysieren die Substantiierer den Text, sie stellen also innerhalb des Analysesystems die einzigen auf dem Text arbeitenden Komponenten dar. Durch Abgleich der im Text gefundenen Worte mit dem Lexikon erhalten die Substantiierer eine umfangreiche Information über die Worte, die sie zur Unterstützung ihrer Arbeit benutzen. Mittels der lexical views wird der Analysevorgang der Substantiierer bedeutend

vereinfacht, da sie sich auf eingeschränkte Teile eines umfangreichen Lexikons beziehen können.

Im Gegensatz zu den klassischen Ansätzen wird von ihnen aber keine umfassende Syntaxanalyse oder vollständige Untersuchung eines Eingabesatzes erwartet. Durch die Kontrolle des Predictors können sie gezielt auf Ausschnitten des Textes eingesetzt werden. Eine Einteilung der Substantiierer kann zunächst nach mehreren Kriterien erfolgen: Welche Aufgabe der *Substantiierung* oder des *Beweisens* haben sie zu lösen, d.h. wie sieht die Erwartung aus? Als zweites Kriterium gilt die Art ihrer Antwort, ist es eine Instantiierung einer Erwartung oder eine Klassifizierung des Dokumentes? Von weiterem Interesse ist die Vorgehensweise der Substantiierer, die oft von ihrer Aufgabe determiniert wird. Sie kann demnach instantiierend, klassifizierend oder im Fall der Regelanwendung, inferierend sein. Zu der klassifizierenden Gruppe zählen vor allem die Substantiierer, die sich auf der Layout- oder Logikebene des Dokumentes bewegen. Eine Ausnahme stellen dabei statistische Klassifizierer dar, die ihre Bewertung aufgrund des Textes fällen. Instantiierende Substantiierer werden oft durch bekannte Parseverfahren der klassischen Syntaxanalyse konstruiert. Die Verwendung des Begriffes inferierender Substantiierer ist nicht ganz korrekt, da es sich um eine Aufgabe des Predictors und nicht um eine eigenständige Komponente handelt. Ihre Beschreibung findet sich daher auch im entsprechenden Kapitel über die Regelanwendung des Predictors.

Die Klassifikation der Substantiierer nach der zu lösenden Aufgabe erlaubt es in den Nachrichtentypen dem Predictor mehrere Substantiierer durch die Klassenangabe als Werkzeuge zur Auswahl zu geben, die dann entsprechend der Qualität der zu beweisenden Erwartung eingesetzt werden. Die folgende Aufzählung stellt einige Substantiierer vor, die für die Analyse durch den Predictor wünschenswert sind.

Struktur-Substantiierer

Für die Anforderungen an das Layout eines Dokumentes oder einzelne seiner Teile werden *Layout-* oder *Strukturs*ubstantiierer erwartet. Sie werden benötigt, um charakteristische Strukturelemente von Layout oder Logik zu finden, die Teil einer Erwartung des Predictors sein können. Mit der Layoutinformation kann eine Zuordnung zu einer logischen Dokumentklasse und damit eine Einordnung des Dokumentes getroffen werden. Diese Substantiierer erfüllen also eine klassifizierende Funktion. Das Dokument kann so einem bestimmten Typ, etwa dem einer Bestellung, zugeordnet werden. Im Umfeld der Dokumentanalyse in ALV (siehe [Dengel 91])ist diese Anforderung durch die Strukturanalyse mit Segmentierung und Logical Labeling bereits erfüllt. Damit kann direkt auf logische Objekte oder Layoutobjekte zugegriffen werden.

Schlüsselwort-Substantiierer

Die Erwartungen, die der Predictor generiert, geben als Constraint möglicher Antworten oft eine lexical view an, mit der eine geringe Auswahl von Worten des Lexikons bezeichnet wird. Beispiele dafür sind einschränkende Sichten wie `LV-Besteller` oder `LV-Produkt`. Mit einem Schlüsselwort-Substantiierer können diese schnell i Text gefunden werden. Auch zur

Klassifizierung kann dieser Substantiierer benutzt werden, indem der Text nach charakteristischen Worten einer Dokumentklasse durchsucht wird. Eine Klassifizierung eines Dokumentes als Bestellung läßt sich dadurch rechtfertigen, daß das Verb "bestellen" und einige Produktnamen im Text auftauchen. Da eine einfache Schlüsselwortsuche dadurch, daß sie den Kontext vernachlässigt, nicht als sichere Quelle dienen kann, wird sie zu einem Schlüssel-Substantiierer erweitert, der in der Suche den Kontext partiell berücksichtigt. Statt der Zeichenkette wird nach einem Muster (ähnlich einem regulären Ausdruck) gesucht. Die Beschreibung einer möglichen Muster-Erkennungs-Sprache findet sich in [Hayes 85a]. Mit dem Beispielmuster `(LV-bestellen +V (&skip 3 (LV-Firma !! LV-Produkt !! LV-Aktie)))` wird ein Verb, das u.a. die Bedeutung `LV-bestellen` hat, nur dann gefunden, wenn darauf ein Wort der Klassen `LV-Firma`, `LV-Produkt` oder `LV-Aktie` mit einem Abstand von nicht mehr als 3 Worten folgt. Damit wird eine Klassifizierung eines Dokumentes durch den Satz „Wir bestellen Ihnen schöne Grüße" als Bestellung erfolgreich vermieden.

Phrasen-Substantiierer

Typische Phrasen eines Geschäftsbriefes wie das übliche „mit freundlichen Grüßen" haben keinen Einfluß auf die Bedeutung des gesamten Dokumentes. Innerhalb des Briefes können aber wichtige Aussagen durch Phrasen umschrieben sein. Statt des Verbs "bestellen" taucht die Phrase "schicken Sie uns" oder eine noch schwerer verständliche Umschreibung eines Bestellens auf. Die Rolle des Phrasen-Substantiierers (siehe [Becker 75]) stellt hier eine Erweiterung der Schlüsselwort-Substantiierer dar, der diese umschreibenden Phrasen, die als lexical views definiert sind, aufspürt. Damit die gesuchten Phrasen und Floskeln innerhalb des Kontextes bewertet werden, wird auch die Phrasen-Suche erweitert und mit der Angabe eines Musters, statt nur der lexical view aufgerufen. Der Phrasen-Substantiierer kann wie der Schlüsselwort-Substantiierer als klassifizierender Substantiierer eingesetzt werden[3].

Muster-Klassifikator

Eine zweite verläßliche Möglichkeit der Klassifizierung wird durch eine Muster-Klassifizierung ([Hayes 88]) erreicht. Um die Einteilung zu einer Klasse zu verifizieren, benutzt das System ein *pattern-matching*-Verfahren, das den Kontext in gewissen Grenzen berücksichtigt. Die Muster werden dabei mit einer Muster-Beschreibungssprache definiert. Dabei werden durch die Muster eine Menge von Phrasen oder Worten angegeben, die mit einem bestimmten Konzept verbunden werden. Für das Konzept "bestellen" gibt das entsprechende Muster die Phrasen an, die in Dokumentklassen auftauchen können, die zu diesem Konzept gehören. Durch die Beachtung des Kontextes in den Mustern, mit denen ein Wort gesucht wird, geht die Leistungsfähigkeit dieses Verfahrens weit über die einer einfachen Schlüsselwortsuche hinaus.

---

[3] Im Rahmen einer Projektarbeit wurde ein Phrasen/Schlüsselwortsubstantiierer, der mit dieser Musterbeschreibungssprache arbeitet, implementiert ([Schmidt 93]).

Mit der Hypothesenbildung und der Verifikation wird die Klassifikation des Dokumentes in zwei Phasen aufgeteilt. In der ersten Phase, der Hypothesenbildung, wählt das System alle Dokumentklassen aus, in die der Text aufgrund seiner Worte und Phrasen fallen kann. Dazu wird ein *pattern-matching* mit einer konstanten Menge von Mustern benutzt. Die Muster werden alle auf den Text angewendet und nach der Zahl der Treffer bewertet. In den Hypothesenregeln werden für jede Klasse Schwellwerte angegeben, die auf Erfahrungen basieren. Der Text kann nur den Klassen zugeordnet werden, für die die Trefferzahl des Musters den Schwellwert übersteigt. Nach dieser Phase steht für das Dokument fest, für welche Dokumentklassen eine Hypothese gebildet werden kann. In der Verifikationsphase werden zusätzliche Hinweise gesucht, die eine Zuordnung zu einem Dokumenttyp bestätigen können, oder aber die fälschliche Zuordnung erkennen lassen, weil Worte falsch interpretiert wurden. Die Verifikation benutzt dazu spezielle Muster in Abhängigkeit von den Hypothesen, die über den Text in der ersten Phase aufgestellt wurden. Die Verifikationsregeln, die von einer Hypothese zu den Mustern für die Verifikationsphase führen, werden wie die Hypothesenregeln in einer Wissensbasis dargestellt. Mit diesem System zur Klassifizierung lassen sich sehr gute Ergebnisse erzielen, vorausgesetzt die Muster zur Hypothesenüberprüfung, die Schwellwertregeln und die Verifikationsregeln sind genau an die Art der erwarteten Texte angepaßt. Das erfordert, daß die Texte der Domäne eine gewisse Strukturierung aufweisen.

Statistischer Klassifikator

Im Gegensatz zu den bisher genannten Substantiierern, die, klassifizierend eingesetzt, sehr stark vom Vorkommen einzelner Worte oder Phrasen abhängen, wählt dieser Klassifikator einen anderen Weg. Basierend auf einer statistischen Analyse vieler Briefdokumente werden bestimmte bedeutungstragende Worte des vorliegenden Dokumentes extrahiert und gewichtet (siehe [Dittrich 92], [Hoch & Dengel 93]). Daraus kann eine Klassifizierung des Dokumentes berechnet werden. Ein klassifizierender Substantiierer stellt für den Predictor eine große Hilfe dar. Durch die Einordnung des Dokumentes in eine Klasse kann der Predictor starke Erwartungen aufbauen und so die Analyse insbesondere in der Startphase beschleunigen.

Fuzzy-Parser

Der Sinn einer durch den Predictor gesteuerten Analyse liegt darin, nicht jedes Wort des Textes zu lesen, sondern den Text gezielt zu überfliegen. Zudem sollen Fehler der Eingabe nur geringen Einfluß auf das Ergebnis der Analyse haben. Für diese Zwecke ist ein *Fuzzy-Parser* besonders geeignet. Gewöhnlich wird *fuzzy parsing* zur Erkennung fehlerbehafteter Texte eingesetzt, um wenigstens einen Rest der Satzinformation mitzubekommen. Der erwartungsgesteuerten Analyse kommt er durch das Überspringen für den konkreten Bedarf unwichtiger Worte entgegen. Der *Fuzzy-Parser* benötigt nur eine rudimentäre Grammatik, die ausreicht, um die Subjekt-Verb-Objekt-Struktur eines Satzes zu erkennen, womit die wesentlichen Anforderungen des Predictors erfüllt sind. Kann dieser Substantiierer die Antwort als Conceptual Dependency-Form zurückliefern, also in der Struktur, mit der die

Nachrichtentypen aufgebaut und die Erwartungen beschrieben sind, wird die Anzahl der wiederholten Substantiiereraufrufe vermindert.

Insel-Parser

Das gleiche Argument der überfliegenden Analyse des Textes gilt auch für die Anforderung eines *Insel-Parsers* ([Stock 89]). Die Erwartungen des Predictors erfordern oft, daß die syntaktische Analyse an einem Punkt beginnt, der semantisch zwar von zentralem Interesse ist, aber für ein klassisches *parsen* von links nach rechts nicht geeignet ist, weil er inmitten einer Satzstruktur liegt. Wenn z.B. ein erster Schlüsselwort-Substantiierer das bedeutsame Verb "bestellen" gefunden hat, wird der Predictor als nächste Schritte das Subjekt und das Objekt des Verbs anfordern. Die Analyse kann dabei nur an dem Verb "bestellen" starten und muß sich von dieser Insel zu den gesuchten anderen vorarbeiten. Auch ein Insel-Parser kann eine gesamte *Conceptual Dependency*-Form als Antwort angeben und so der Aufgabe des Predictors entgegenkommen.

Andere Substantiierer

Die bisher vorgestellten Substantiierer wurden mit einer Motivation angegeben, die ihre Anwendung in einem erwartungsgesteuerten System begründet. Sie alle - bis auf den statistischen oder Muster-Klassifikator - stellen eigentlich unvollständige *Parser* dar, die gezielt auf diese Anwendung konzipiert sind. Stattdessen könnten aber auch Substantiierer benutzt werden, die mit einer vollständigen Grammatik arbeiten und den gesamten Text satzweise syntaktisch vollständig analysieren.

Als Beipielformalismen seien nur ATN-Grammatiken (*augmented transition network grammars*), DCGs (*definite clause grammars*) und HPSGs (*head driven phrase structure grammars*) genannt (siehe [Gores 90]). Unter Ausnutzung der semantischen Fähigkeiten dieser Formalismen kann eine Struktur aufgebaut werden, die reich an Informationen und für den Predictor leicht in instantiierte Erwartungen durch Nachrichtentypen umsetzbar ist. Eine ebenfalls interessante Wahl stellt die Analyse durch eine DRS (Diskursrepräsentationsstruktur, siehe [Kamp 88]) dar, die dem Text als Ganzem eine syntaktisch/semantische Struktur zuweist.

Die Vorteile einer Verwendung solcher Substantiierer liegen auf der Hand: mit der Struktur eines (semantisch attributierten) Syntaxbaumes kann eine Substantiierung gezielter erfolgen. Das Überfliegen des Textes, das unter Umständen auch ein Überlesen bedeuten kann, wird vermieden. Die Semantik des Dokumentes, im Sinne der instantiierten Erwartungen des Predictors, kann dann leichter aufgebaut werden. Solange der Eingabetext keine oder nur sehr "unbedeutende" Fehler aufweist, sind auch solche Verfahren begrenzt einsetzbar. Für sprachlich schwierige oder zweifelhafte Konstrukte hingegen ist es nicht empfehlenswert. Daher sind die eben vorgestellten alternativen Substantiierer für die erwartungsgesteuerte Analyse meist nicht so geeignet.

Schnittstelle der Substantiierer zum Predictor

Nachdem die Substantiierer nach ihrer Leistungsfähigkeit bezüglich der Aufgaben eingeordnet wurden, muß noch das erwartete Ausgabeverhalten als andere Seite der Schnittstelle zum Predictor erläutert werden. Auf dieser Schnittstelle werden die Antworten der Substantiierer auf die Erwartungen des Predictors transportiert. Die Verschiedenheit der Substantiierer bedingt auch eine unterschiedliche Qualität der Antworten. Das Ergebnis einer Klassifizierung liefert Hinweise auf den Nachrichtentyp des vorliegenden Dokumentes, dies kann eine Zeichenkette oder eine Liste sein. Die Antworten der Schlüsselwort-Substantiierer können als ja/nein-Antwort erfolgen, oder aber das gefundene Wort bzw. die gefundene Phrase zurückliefern.

Der Predictor fordert von allen Substantiierern außer den Klassifikatoren, daß zumindest das Wort oder die Phrase, die den Slot gefüllt oder auf die das Suchmuster gepaßt hat, als Antwort zurückgegeben wird. Diese Antwort muß als differenzierte Information die Zeichenkette im Eingabetext enthalten, die Wortposition und den Verweis auf den Lexikoneintrag. Damit kann der Predictor auf das gesamte Lexikonwissen des *Slot*-Füllers zugreifen. Die erweiterte Forderung richtet sich an die geschickteren Substantiierer, die ein größeres syntaktisches oder gar semantisches Wissen haben. Diese sollen mit *Conceptual Dependency*-Formen antworten. Diese können so umfangreich sein, daß damit ein gesamtes Nachrichtenelement als Antwort zurückgeliefert wird.

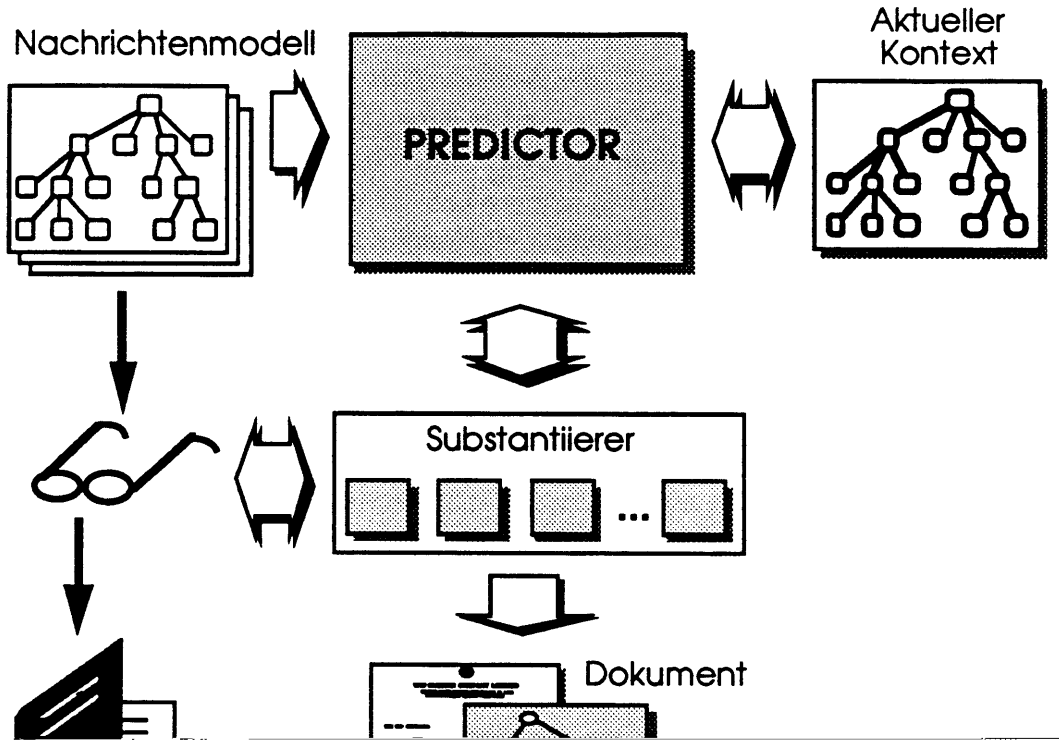
Unter bestimmten Umständen kann eine Substantiiererantwort mehrdeutig ausfallen, z.B. wenn durch eine fehlerhafte Texterkennung zu mehreren möglichen Alternativen ("*H*aus","*M*aus","*L*aus") führte und von diesen mehrere einen erwarteten Slot füllen können. In diesem Fall muß der Predictor alle passenden Alternativen akzeptieren und eintragen. Die Auswahl der wahrscheinlichsten Alternative kann durch weitere aus dem Text gewonnene Information zu einem späteren Zeitpunkt möglich sein.

Zusammenfassung

Mit den verschiedenen Arten von Substantiierern, die entsprechend ihrer Leistungsfähigkeit und ihrem Ein-/Ausgabeverhalten klassifiziert werden, stehen dem Predictor eine Fülle an Werkzeugen zur Verfügung, die er zum Beweis seiner Erwartungshaltungen benutzen kann. Für alle Elemente der Nachrichtentypen, also alle Erwartungshaltungen, wird angegeben, welche Substantiierer-Leistung vonnöten ist, um für dieses Element einen passenden Füller im Text zu finden. Welcher der fähigen Substantiierer damit beauftragt wird ist Aufgabe des Predictors.

2.4 Prinzipieller Ablauf

Der prinzipielle Ablauf der erwartungsgesteuerten Analyse soll anhand der folgenden Abbildung 3 verdeutlicht werden.


Nachrichtenmodell
PREDICTOR
Aktueller Kontext
Substantiierer
Dokument

Textkontext die korrekte Repräsentation des vorliegenden Dokumentes als Instanz eines oder mehrerer Nachrichtentypen enthält. Der Predictor füllt den Kontext, indem er für die Erwartung geeignete Substantiierer auswählt und mit der Bestätigung einer Erwartung aufruft. Das Ergebnis der Anfrage erweitert den aktuellen Kontext. Innerhalb des Nachrichtenmodells sind die gesamten Erwartungen über die verschiedenen Typen von Nachrichten kodiert. Hier wird vorgeschrieben, welche wichtigen Bestandteile ein Brief einer bestimmten Klasse enthält und aus welchen Worten diese Bestandteile bestehen dürfen. Um die Art der Worte einzugrenzen, werden in den Nachrichtentypen lexical views als Constraints angegeben, die die Sicht auf das Lexikon einengen. Wenn die Substantiierer mit einer Bestätigungsaufgabe betraut sind, benutzen sie das Lexikon, um umfangreiche Informationen über gefundene Worte zu erhalten.

In der Analyseschleife stellt der Predictor als zentrale Instanz aus den Informationen des aktuellen Kontextes und dem Nachrichtenmodell eine gezielte Erwartungshaltung auf. Zu deren Bestätigung zieht er einen der Substantiierer heran, wobei eine Vorauswahl ebenfalls im Nachrichtenmodell festgelegt ist. Die Substantiierer, die als einzige Elemente auf dem Text arbeiten, sind spezialisierte Parser verschiedener Leistungsklassen, die die Erwartungen des Predictors an den Text unter Zuhilfenahme des Lexikons und der lexical views erfüllen. Die erfolgreiche Antwort eines Substantiierers wird als Erweiterung des aktuellen Kontextes eingetragen. Mit dem erweiterten Wissen beginnt eine neue Phase der Erwartungsgenerierung, solange bis alle Erwartungen befriedigt sind.

Auf diese Weise wird entsprechend der Vorhersagen des Predictors, der Nachrichtentypen des Nachrichtenmodells, den Substantiierern und dem Lexikon die Information des Dokumentes gefunden, wenn die Erwartungshaltung richtig war. Das Konzept des ausgewählten Nachrichtentypen wird schrittweise als aktueller Kontext instantiiert und kann anschließend interpretiert werden.

3 Der Predictor

Der Predictor ist die zentrale Komponente der erwartungsgesteuerten Analyse. Er steuert das Vorgehen zur Informationsextraktion des Textes in Abhängigkeit vom bereits identifizierten Text und den generierten Erwartungen. Dabei legen Datenstruktur und Inhalte der Nachrichtentypen und die im Text gefundenen Elemente den genauen Ablauf fest. Hier soll noch einmal kurz das Prinzip der erwartungsgesteuerten Analyse vorgestellt werden, um dann die Arbeitsweise des Predictors en détail zu beschreiben.

Das Modell der erwartungsgesteuerten Textanalyse

Die erwartungsgesteuerte Analyse eines Dokumentes ist durch folgende, sich zyklisch wiederholende Aktionen gekennzeichnet: Der Predictor wählt Erwartungen aus der Gesamtmenge der Nachrichtentypen aus. Diese Erwartungen stellen Vermutungen über die im Dokument enthaltenen Informationen auf. Die Substantiierer versuchen, beauftragt durch den Predictor, die Verifikation dieser Erwartungen.

Innerhalb dieser Schleife wird durch die Ergebnisse der bisherigen Analyse und Unterstützung des Nachrichtenmodells ein aktueller Kontext aufgebaut. Der aktuelle Kontext umfaßt den Stand der Analyse sowohl durch die Menge der Erwartungen (Erwartungskontext) und die in Erwartungen instantiierte Textinformation (Textkontext). Die Erwartungen sind dabei Elemente des Nachrichtenmodells, z.B. Nachrichtentypen oder Nachrichtenelemente. Der Textkontext besteht aus erfüllten Erwartungen, also an Worte des Textes gebundene Erwartungen.

Die neuen Erkenntnisse einer Iteration fließen in den aktuellen Kontext als weitere Information ein, so daß die nachfolgenden Schritte der Analyse davon profitieren können. Dabei stellt der Beginn der Analyse für den Predictor die schwierige Situation dar, als Vorwissen ausschließlich das Nachrichtenmodell als die Menge aller möglichen Erwartungen zu haben, aber keine Information aus dem Text. Mit dieser unbeschränkten Informationsfülle läßt sich keine Vorhersage sinnvoll vor allen anderen auswählen. Das Übermaß an Information ist damit wertlos und führt zum Dilemma des *Startproblems*: soll vom angestrebten *top-down* Vorgehen abgewichen oder blind nach signifikanten Elementen der Nachrichtentypen gesucht werden? Die Realisierung einer naiven bottom-up Lösung wird

im Zusammenhang mit der Startdiskriminierung erläutert. Sie besteht aus einer Aufforderung an einen Substantiierer, irgendetwas aus dem Text zu liefern[4]. Mit der Antwort soll die Analyse dann erwartungsgesteuert fortgeführt werden. Dabei entscheidet die Qualität der „irgendetwas"-Antwort darüber, wie schnell die Startphase überwunden werden kann. Die zweite Lösung der blinden Suche nutzt die Information in den Nachrichtentypen mehr, indem nach den ausgezeichneten Schlüsselelementen aller Nachrichtentypen gesucht wird. Diese Aufgabe wird u.a. von den Schlüsselwort-Substantiierern, die in Kapitel 2 angesprochen wurden, erledigt. Kann von einer stark eingeschränkten Domäne mit wenig Schlüsselworten ausgegangen werden, ist dieser wenig zielstrebige Weg mit schnellen Suchverfahren denkbar. Allerdings ergeben sich damit auch neue Probleme: Das Suchen nach Schlüsselelementen ist nicht verläßlich, da ein gefundener Schlüssel in einem die Bedeutung verfälschenden Kontext enthalten sein kann, z.B. innerhalb von Anführungszeichen. Außerdem ist die Suche nur von einer unzureichenden Erwartungshaltung motiviert.

Ein anderer Weg zur Lösung des Startproblems bietet als besonderer Substantiierer ein *Klassifikator* (siehe [Dittrich 92], [Hoch & Dengel 93]), der Hinweise auf den im Text vorliegenden Nachrichtentyp liefert. In einer Liste werden für die in Frage kommenden Nachrichtentypen die Wahrscheinlichkeiten angegeben. Dieser ausgezeichnete Substantiierer klassifiziert das Dokument nach statistischen Methoden, ein Verfahren, das in seiner Verläßlichkeit der Schlüsselsuche überlegen ist. Damit wird die Diskriminierungsphase, also das Verfahren zur Festlegung auf einen Nachrichtentyp, abgekürzt. Der gewonnene Kontext, nämlich die Zuordnung zu einem Nachrichtentyp, erlaubt eine weniger blinde Suche oder schränkt die Vorhersagenmenge so ein, daß Erwartungen generiert werden können.

Weitere Alternativen zur Lösung des Startproblems sind die Verfahren der iterierenden Diskriminierung als naive Diskriminierung oder durch sogenannte Diskriminierungsbäume (siehe Kapitel 3.3). Die Aufgabe des Predictors besteht darin, aus dem sukzessiv gewonnenen Kontext und dem Nachrichtenmodell gezielt neue Vorhersagen zu generieren. Dazu muß die Steuerinformation in den Nachrichtentypen interpretiert werden, um die Vorhersagenmenge einzuschränken und geeignete Substantiierer auszuwählen. Die Antwort der Substantiierer als Erweiterung des Kontextes beeinflußt dann das weitere Vorgehen.

Überblick

Die Analyse des Predictors gliedert sich in folgende 3 Phasen:

- Startphase: Der Predictor hat keinerlei Vorwissen aus dem Text (leerer Textkontext) und einen Erwartungskontext, der aus dem gesamten Nachrichtenmodell besteht. Der Predictor baut einen eingeschränkten Erwartungskontext auf, um eine systolische (die

- Diskriminierungsphase (Systole): Der Predictor hat Vorwissen durch den aktuellen Kontext (z.B. durch während der Startphase instantiierte Vorhersagen) und die bereits eingeschränkte Erwartungsmenge. Durch Diskriminierungsverfahren wird die Erwartungsmenge auf einen Nachrichtentyp eingeschränkt.
- Instantiierungsphase (Diastole): Alle Erwartungen des ausgewählten Nachrichtenelementes werden aus dem Text als Erweiterung des Textkontextes instantiiert. Dazu muß zunächst die Menge der Erwartungen erweitert werden.

Dabei wiederholen sich die systolische (also die Vorhersagenmenge einengende) und diastolische (die Vorhersagenmenge erweiternde und erfüllende) Phase solange, bis die Analyse beendet ist.

Die Kontrolle während der Textanalyse, die durch die Einbettung in das Gesamtsystem der Dokumentanalyse in ALV von Vorarbeiten anderer Komponenten profitieren kann, liegt

Textanalyse als auch in der darauffolgenden Diskriminierungsphase angewendet. Dabei werden für die Startphase sowohl Verfahren eingesetzt, die nur hier auftauchen, als auch Spezialisierungen von Vorgehensweisen, die sich auch in der Diskriminierungsphase wiederfinden.

Zunächst werden jedoch die Datenstrukturen vorgestellt, die zur Diskriminierung verwendet werden. Sie werden aus dem Nachrichtenmodell gebildet, d.h. aus der Gesamtheit der dort definierten Nachrichtentypen.

3.1.1 Datenstrukturen der Diskriminierung

Bevor im nachfolgenden Kapitel die denkbaren Verfahren diskutiert werden, um mit dem Wissen des Nachrichtenmodells eine effiziente Analyse betreiben zu können, muß dieses Wissen den Bedürfnissen des Predictors entsprechend strukturiert werden. Die Möglichkeiten des Zugriffes und der Organisation der Nachrichtentypen hat entscheidenden Einfluß auf die anwendbaren Verfahren und den ihnen beschiedenen Erfolg.

Die Definition der Nachrichtentypen im Nachrichtenmodell ([Gores & Bleisinger 92]) wurde durch eine hierarchische Klassenstruktur festgelegt. Dabei stellen die Nachrichtenelemente einen Vorrat von Bausteinen bereit, aus dem die Nachrichtentypen aufgebaut werden. Dieser Aufbau läßt sich als eine Baumstruktur, die dahingehend leicht erweitert wurde, daß mehrere Wurzeln erlaubt sind, visualisieren (siehe Abbildung 4)

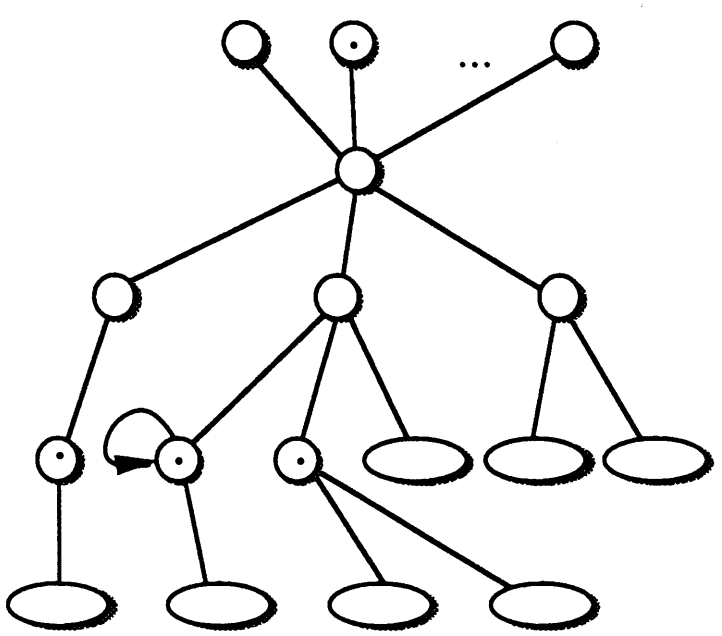


Abbildung 4: Der Nachrichtenbaum gemäß der Definition

Hierbei stellen die Knoten einzelne Nachrichtenelemente dar, wobei die wichtigen[6] durch ein "•" markiert sind. Die Beschreibung eines Nachrichtentyps ist am Blatt eines Pfades durch den Nachrichtenbaum vollständig. Diese Organisationsform des Nachrichtenbaums kann in allen Phasen der Analyse durch den Predictor benutzt werden. Diese Struktur des Nachrichtenbaumes wird aus dem Nachrichtenmodell konstruiert.

Die Nachrichtentypen können aber auch auf andere Weise organisiert werden, z.B. als Listen oder als spezielle Bäume. Dabei muß die realisierte Implementierung des Zugriffs auf die Nachrichtentypen für den Predictor nicht transparent sein. Es genügt, wenn die benötigten Zugriffsfunktionen bereitstehen. In allen Varianten der Organisation ist es nötig, eine *prototypische Instantiierung* des Nachrichtenmodells zu erzeugen. Nur so können die Inhalte der Nachrichtentypen, also insbesondere die Steuerungseinheiten, ausgewertet werden.

Listenorganisation

Die prototypischen Instanzen aller Nachrichtentypen werden in einer Liste gesammelt, jedes Element besteht aus den Instanzen der Nachrichtenelemente eines Nachrichtentyps (siehe Abbildung 5).

Abbildung 5: Die Nachrichtentypen in der Listensicht

Die einzigen Zugriffsfunktionen auf die Liste, `car` und `cdr`, erlauben lediglich einen sequentiellen Zugriff. Um z.B. alle Nachrichtentypen mit einem bestimmten Nachrichtenelement auszufiltern, müssen alle Listenelemente durchsucht werden. Da die Nachrichtentypen eine mehrstufige Datenstruktur haben, stellt die Suche nach einem speziellen Kriterium, wie dem Slotwert eines Nachrichtenelementes, durch den geschachtelten Zugriff eine die Performanz des Systems gefährdende Belastung dar. Der Zugriff zum ausfiltern aller Nachrichtentypen mit einem bestimmten semantischen Constraint des `action-slot` erfordert:

- den sequentiellen Zugriff auf alle Listenelemente (Nachrichtentypen)
- den Zugriff auf alle Nachrichtenelemente durch die entsprechenden `accessor`-Funktion
- den Test auf Vorhandensein eines `action-slots`
- den Slot-Zugriff durch den Slot-`accessor`
- den Aspektzugriff durch den Aspekt-`accessor`.

---

6 "wichtig" bezieht sich hierbei auf den `importance`-Eintrag des Nachrichtenelementes und der Einträge zur expliziten Aktivierung (`explicit-reference`) sowie zur elementinduzierten Aktivierung (`element-induced-activation`)

Die Listenorganisation bereitet geringe statische Kosten, d.h. die zusätzliche Listenstruktur der Instanzen der Nachrichtentypen läßt sich leicht erzeugen und benötigt wenig zusätzlichen Speicherplatz. In Anwendung und Zugriff ist sie beschwerlich und ineffizient, hohe dynamische Kosten sind die Folge.

Baumorganisation

Die prototypische Instantiierung der Nachrichtentypen als Baumstruktur der Definition wie in Abbildung 4 bietet wesentliche Vorteile. Das Ausfiltern bestimmter Nachrichtentypen bedeutet in den günstigsten Fällen lediglich das Beschneiden des Baumes. Der Aufwand zur Erstellung der Baumorganisation ist aber aufgrund der Definition des Nachrichtenmodells nicht unerheblich. Dieser muß jedoch nur einmal betrieben werden, wenn sich die Definition des Nachrichtenmodells ändert. Diese Baumorganisation wird daher favorisiert verwendet.

Die Baumstruktur beschleunigt, erst einmal aufgebaut, den Zugriff und das Ausfiltern von Nachrichtentypen, also die häufigste Anwendung. Die dynamischen Kosten werden damit klein gehalten. Für eine Domäne, die durch eine große Anzahl an Nachrichtentypen und Nachrichtenelementen beschrieben werden muß, sollte der Baum beschnitten werden. Der Maßstab des Beschneidens sind dabei die wichtigen Elemente der Nachrichtentypen entsprechend der `importance`-Einträge im Nachrichtenmodell und die Steuerinformationen der expliziten sowie elementinduzierten Aktivierung. Abbildung 6 zeigt den beschnittenen Baum, der durch Ausblenden der unwichtigen Nachrichtenelemente des Baumes aus Abbildung 4 entstanden ist. Dieser beschnittene Nachrichtenbaum wird im vorgestellten System in der Start- und Diskriminierungsphase benutzt.

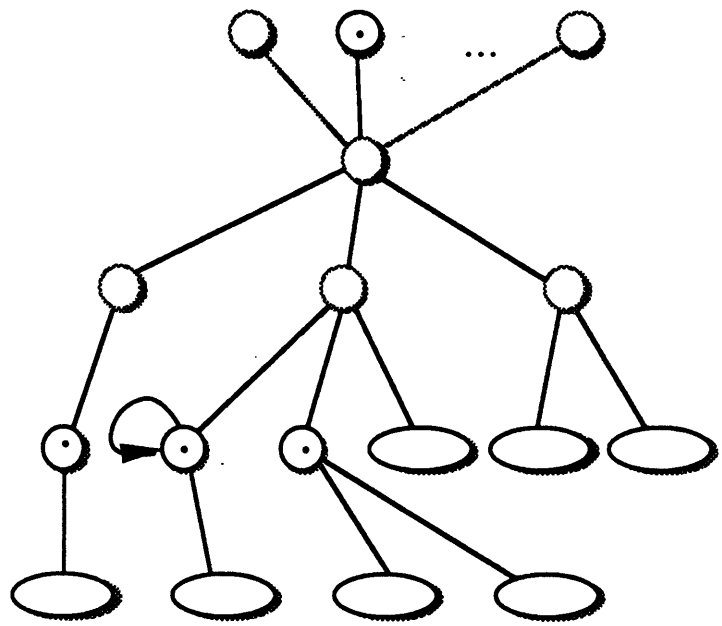


Abbildung 6: Der beschnittene Nachrichtenbaum

Als Spezialisierung dazu werden die *Diskriminierungsbäume* konstruiert. Sie repräsentieren nicht mehr die gesamte Struktur der Nachrichtentypen, sondern nur die zur Diskriminierung nötigen Informationen, also der Einträge der `element-induced-activation`-Slots und der `explicit-reference`-Slots. In Abbildung 7 ist als Beispiel ein Diskriminierungsbaum der aktiven CD-Formen der Nachrichtenelemente skizziert.

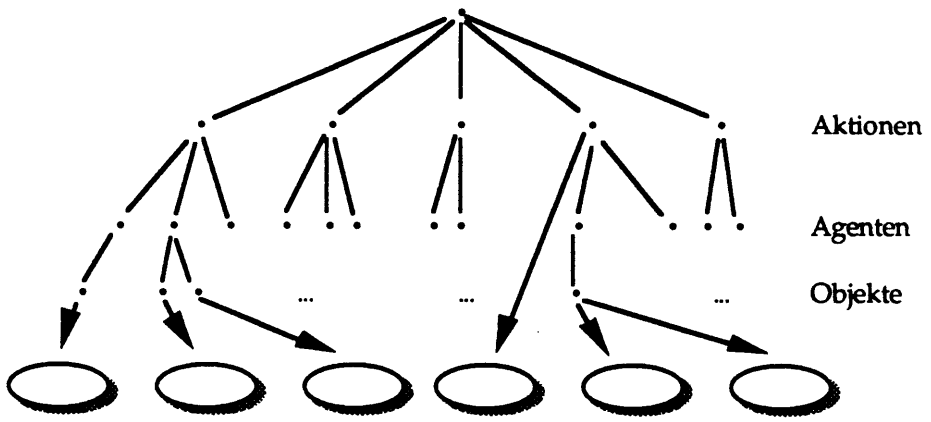


Abbildung 7: Der Diskriminierungsbaum für Aktionen

Über maximal 3 Stufen führt die Diskriminierung vom (leeren) Startknoten zu einem Nachrichtentyp, oder einer geringen Auswahl an Nachrichtentypen. Für Zustände, also statische CD-Formen, wird ebenfalls ein Diskriminierungsbaum angegeben. Die Zustandsdiskriminierung läuft über das Objekt, den Zustand und dessen Werte. Die Diskriminierungsbäume werden im vorgestellten System in der Startphase benutzt.

Der statische Aufwand zur Konstruktion der Diskriminierungsbäume besteht darin, die Informationseinheiten zur elementinduzierten Aktivierung auszuwerten und aus diesen und den korrespondierenden Nachrichtenelementen die Diskriminierungsbäume aufzubauen. Die Reihenfolge der Knoten der Diskriminierungsbäume - im Beispiel Wurzel, Aktionen, Agenten und Objekte - muß dabei der Art der Konzeptualisierung angepaßt sein.

3.1.2 Verfahren der Diskriminierung

Das wesentliche Ziel der Diskriminierung besteht darin, den Erwartungskontext soweit einzuschränken, daß vom Predictor gezielt Vorhersagen aufgebaut und an die Substantiierer zur Verifikation übergeben werden können. Dabei besteht die Hauptaufgabe darin, aus den im Nachrichtenmodell vorgegebenen möglichen Erwartungen eine Diskriminante zu berechnen.

Die Diskriminante

Die Diskriminante wird als Vorhersage aus den konkurrierenden Slots der Nachrichtenelemente konstruiert, indem die widersprechenden Constraints zu einem neuen vereinigt werden. Innerhalb der Phase der iterierenden Diskriminierung auf einem Erwartungskontext, der nur aus den Schlüsseln, also den wichtigsten Elementen der Nachrichtentypen besteht, erfolgt die Wahl der Diskriminante nach folgenden Richtlinien:

- Entscheidungskraft: Wie selektiv kann eine Antwort eines Substantiierers auf die Diskriminante als Anfrage sein?
- Kosten: Wie teuer ist die Substantiierung der Anfrage?

Wenn der Erwartungskontext aus allen Nachrichtenelementen eines Nachrichtentyps und den Schlüsseln der impliziten Aktivierung besteht, muß die Diskriminante nach dem zusätzlichen Kriterium

- Qualität: Ist der Slot als Diskriminante sinnvoll?

bewertet werden. Damit wird vermieden, daß weniger wichtige Einträge zur Diskriminierung herangezogen werden, anstatt zuerst entscheidende zu betrachten, wie z.B. Aktionen.

Hat der Predictor durch die Diskriminierung den Kontext soweit eingeengt, daß er aus einem aktiven Nachrichtentyp und eventuell der Erwartung impliziter Aktivierungen besteht, so beginnt die diastolische Phase. Dieser *bewiesene Kontext* stellt für die iterierende Diskriminierung eine Invariante dar, der schrittweise instantiiert und zu dem nach einer erfolgreichen Diskriminierung wieder zur Auswahl des nächsten Kandidaten zurückgekehrt wird.

Listenorganisation

Mit der quasi-flachen Listenstruktur muß sich die Diskriminierung explizit um jedes Element eines jeden Nachrichtentyps bemühen. Was das bedeutet, kann am Beispiel der Startdiskriminierung leicht deutlich gemacht werden:

Die Liste der Nachrichtenelemente eines jeden Nachrichtentyps muß auf die Elemente beschränkt werden, die in den entsprechenden Steuerungseinträgen ihres Nachrichtentyps als Schlüssel der expliziten oder elementinduzierten Referenz markiert wurden. Nur diese Elemente werden für die Startdiskriminierung verwendet. Dazu müssen für jeden Nachrichtentyp die Slots dieser beiden Aktivierungsformen betrachtet werden, um die Elemente, die nicht in dieser Liste enthalten sind, auszufiltern. Für eine große Anzahl an Nachrichtentypen und Nachrichtenelementen kann dieser Aufwand beträchtlich sein. Allerdings kann diese, nur jeweils aus den Schlüsseln der Nachrichtentypen bestehende Liste als Konstante des Nachrichtenmodells definiert werden. Zu einem späteren Zeitpunkt der Diskriminierung, die das Ausfiltern von Nachrichtenelementen mit bestimmten Eigenschaften oder Constraints erfordert, muß die neue Liste dennoch jedesmal berechnet werden.

Zur Diskriminierung innerhalb der quasi-flachen Listenstruktur werden Funktionen zum Ausfiltern von Nachrichtenelementen und Nachrichtentypen aufgrund von Constraints benötigt, die Aussagen über Slots von Nachrichtenelementen oder deren Aspekte machen. Wird ein Nachrichtentyp ausgewählt, so muß die Listenstruktur auf alle Nachrichtenelemente dieses Typs erweitert werden. Diese Vorhersagenmenge dient dann als Vorgabe zur Instantiierungsphase.

Der Aufwand ist gegenüber der unten vorgestellten Baumvariante also bedeutend erhöht, die Diskriminierung in der quasi-flachen Listenstruktur kann nur für eine geringe Anzahl von Nachrichtenelementen und Nachrichtentypen oder, wenn keine Möglichkeit des Baumzugriffs besteht, sinnvoll sein.

Diskriminierung in einer baumähnlichen Struktur

Kann auf die Nachrichtenelemente als Knoten und die Nachrichtentypen als Pfade einer Baumstruktur zugegriffen werden, läßt sich der dynamische Aufwand zur Diskriminierung bedeutend mindern. Ebenso wie für den Fall der quasi-flachen Liste wird eine zweite Struktur angelegt, der in Abbildung 6 bereits skizzierte beschnittene Nachrichtenbaum, der nur die der Diskriminierung förderlichen Nachrichtenelemente, die Elemente zur expliziten und elementinduzierten Aktivierung, enthält.

Damit wird die Funktion des Ausfilterns von Nachrichtenelementen und Nachrichtentypen aufgrund von Constraints durch das Beschneiden von Ästen des Baumes effizient realisiert. Zur völligen Instantiierung des ausgewählten Nachrichtentyps müssen dann die übrigen Elemente, die für die Diskriminierung keine Bedeutung haben, in die Vorhersagenmenge aufgenommen werden. Dazu wird nach Erreichen des Blattknotens des beschnittenen Nachrichtenbaumes, wenn also die Evidenz für einen Nachrichtentyp hinreichend ist, dieser Typ aktiviert.

In der auf die Instantiierungsphase folgenden Bewertung der impliziten Aktivierung, wird die Sicht auf den Baum um die Schlüsselelemente der implizit verbundenen Nachrichtentypen erweitert. Die Diskriminierung bewegt sich dann auf dem Fragment des Diskriminierungsbaums, das dem Multi-Nachrichtentypen entspricht, der mit diesen Nachrichtentypen assoziiert ist.

Diskriminierung mit Diskriminierungsbäumen

Die effizienteste Möglichkeit der Diskriminierung bieten die Diskriminierungsbäume. In wenigen Schritten ist eine Zuordnung des richtigen Nachrichtentyps möglich, bzw. eine Einschränkung, die stark genug ist, um schnell zu einer Entscheidung für einen Typ zu gelangen.

Mit dem in der Abbildung 7 vorgegebenen Diskriminierungsbaum für Aktionen ist eine Diskriminierung über die drei Stufen der Vorhersage einer Aktion, eines Agenten und des Objektes der Aktion möglich. Diskriminierungsbäume realisieren die Idee, daß bestimmte wichtige Elemente durch gezielte Untersuchungen schneller zu einer Entscheidung für einen Nachrichtentyp führen können. Neben dem Diskriminierungsbaum für Aktionen werden auch solche für Zustände (z.B. Adressen, Kunden etc.) eingesetzt.

Variationen der Diskriminierung

Die prinzipiellen Verfahren der Diskriminierung sind besondere Diskriminierungsverfahren der Startphase, Diskriminierungsverfahren unter Benutzung des Nachrichtenmodells (gleich welcher Organisationsform) und Diskriminierung unter Benutzung von Diskriminierungsbäumen. Im folgenden Kapitel wird zunächst die Startphase und die dort verwendeten Diskriminierungsverfahren vorgestellt.

3.2 Die Startphase

Wenn der Predictor die erwartungsgesteuerte Textanalyse startet, liegen zwar durch das Nachrichtenmodell sehr viele Informationen bereit, die aber zur Analyse des Textdokumentes kaum dienlich sind, weil sie nicht eingeschränkt sind. Das Ziel des Predictors muß nun die schnellstmögliche Generierung von Startvorhersagen sein, um einen Erwartungskontext zu schaffen, der neue eingeschränkte Erwartungen durch die Nachrichtentypen bereitstellt. In diesem Kapitel werden verschiedene Ansätze vorgestellt, die zu einer Lösung dieses Problems führen. Dabei stehen die Ansätze als Alternativen zueinander, die sich aber auch ergänzen können. Die Verfahren zur Lösung des Startproblems sind ein Sonderfall der systolischen Phase.

Wurzelknoten des Nachrichtenbaumes als Start

Mit einer baumähnlichen Struktur der Nachrichtentypen kann die Analyse mit der Vorhersage der Informationseinheiten der Wurzeln beginnen. Da alle Elemente entlang des Pfades von den Nachrichtentypen an den Blattknoten auch benötigt werden, wird nichts Überflüssiges instantiiert. Das Ergebnis der Analyse wird fast immer in der Form, in der es instantiiert wird, als Nachrichtenelement des vorliegenden Nachrichtentyps gebraucht. Eventuell müssen die Inhalte mancher Elemente, die an früheren Knoten instantiiert wurden, in andere Knoten kopiert werden, um eine bestimmte Semantik zu erreichen, z.B. `Adresse` in `Firmen-Adr-Adresse`. Dazu müssen die Constraints und Regeln an den Wurzelknoten evalu-

`Adresse` übernommen werden. Dies kann dazu führen, daß Information im Text übersehen

Blattknoten des Nachrichtenbaumes als Start

Die Wahl eines Blattknotens als Start, also die Empfehlung eines Nachrichtentyps, kann nur sinnvoll sein, wenn eine Motivation durch Vorwissen, heuristische oder statistische Indizien vorliegt. Dieses Verfahren setzt also voraus, daß bereits eine klassifizierende Untersuchung des Dokumentes vorgenommen wurde.

Das Vorgehen der Analyse wird dann von den `importance`- und `order`-Constraints, wie sie für den Nachrichtentyp im Modell notiert sind, vorgegeben. Für die Fälle, in denen der falsche Nachrichtentyp erwartet wurde, kann zum einen von den erzeugten partiellen Nachrichtenelementen, also dem Textkontext und dem Erwartungskontext die Fehlerbehandlung gestartet werden. Dazu dienen die instantiierten inneren Knoten und der bewiesene Pfad von einer der Wurzeln zum erwarteten Blattknoten als Hinweise zu alternativen Blättern. Als weitere Hilfestellungen können Verweise auf verwandte Nachrichtentypen benutzt werden, also Querverbindungen im Nachrichtenbaum. Im hier verwendeten Nachrichtenmodell sind allerdings außer den impliziten Referenzen keine weiteren Verbindungen definiert. Durch den Wechsel zu einem anderen Pfad kann das Ignorieren oder Löschen oder, wie beim Erreichen des Blattknotens, das Uminterpretieren von Konzeptualisierungen nötig werden.

Der Vorteil eines Blattknotens als Start liegt in der frühen Fixierung auf einen Nachrichtentyp. Dadurch kann die Steuerinformation vollständig genutzt werden. Allerdings steht und fällt die Auswahl mit der Verläßlichkeit der Quelle der Empfehlung. Ist sie hoch, so bietet der Blattknotenstart ein ideales Verfahren. Im Fehlerfalle muß ein erheblicher Aufwand getrieben werden, um die Analyse fortzuführen. Dies ist aber, im Gegensatz zum Wurzelknotenstart, z.B. durch die Wahl eines inneren Knotens möglich.

Innerer Startknoten des Nachrichtenbaumes

Die Empfehlung eines inneren Knotens muß ebenfalls durch Vorwissen motiviert sein, um die Analyse unterstützen zu können. Dieses Vorwissen muß nicht so stark sein, wie es die Empfehlung eines Blattknotens nötig macht. Leider läßt die Position eines inneren Knoten in der Regel auch nicht die Auswertung der Constraints eines Nachrichtentyp zu, es sei denn, der Pfad vom ausgewählten Knoten verzweigt sich nicht mehr. Durch die untere Baumposition bewegen sich die vorherzusagenden Nachrichtenelemente auf einem Differenzierungsniveau, das den Weg zu einem Blattknoten beschleunigen kann. Im Fehlerfall läßt sich dadurch auch leichter entscheiden, ob dieser Fehler tatsächlich von der falschen Wahl des Starts abhing.

Ohne verläßliche Vorinformationen stellt ein innerer Knoten keine gefällige Wahl dar, da die Nachteile obiger Alternativen verbleiben. Für Nachrichtenbäume geringer Tiefe ähnelt es zu sehr ersterem, für tiefe Bäume muß zuviel Verweisinformation an den inneren Knoten nicht nur angegeben, sondern auch bei der Erstellung des Baumes berechnet werden, um die Fehlerbehandlung zu unterstützen.

Zur Lösung des Startproblemes bietet sich die Wahl von inneren Startknoten dennoch an. Die nötigen Bedingungen zur Rechtfertigung sind nicht so umfangreich wie die für einen Blattknoten, das Ergebnis kann ausreichende Information zur Aktivierung eines

Nachrichtentyps geben. Um die angesprochenen Nachteile zu mindern, ist es wichtig, die Auswahl innerer Knoten auf bestimmte zu beschränken. Dazu bieten sich zunächst die Nachrichtenelemente der Nachrichtentypen an, denen durch die `importance`-Einträge eine hohe Bedeutung zugeordnet ist. Das Fragment des Nachrichtenbaums, das durch Beschneiden der weniger wichtigen Nachrichtenelemente entsteht, ist die Datenstruktur des beschnittenen Nachrichtenbaums (Abbildung 6). Die effizienteste Möglichkeit zur Behandlung der Startphase wird durch Diskriminierungsbäume geboten.

3.2.1 Diskriminierungsverfahren der Startphase

In der Startphase der erwartungsgesteuerten Analyse eines Briefes unter zuhilfenahme des Nachrichtenmodells können prinzipiell zwei Verfahrensweisen unterschieden werden: Von Beginn an wird eine Diskriminierung basierend auf dem Nachrichtenmodell durchgeführt oder es wird zur Beschleunigung dieses Prozesses auf andere Methoden als Vorverarbeitung zurückgegriffen.

Dieser zweite Weg bietet dabei den entscheidenden Vorteil, die Analyse von vornherein nicht auf falschen Pfaden suchen zu lassen, sondern ihr gezielte Hinweise auf den oder die wahrscheinlichsten zu geben. Im Falle eines Vorgehens, das mit keinerlei Vorwissen außer dem Nachrichtenmodell besteht, sprechen wir von einer *naiven Diskriminierung* . Gibt es bereits eine Vorinformation, etwa in der Art einer Empfehlung für einen Nachrichtentyp, kann eine *antizipierende Diskriminierung* durchgeführt werden.

3.2.1.1. Antizipierende Diskriminierung

Der Erfolg einer erwarungsgesteuerten Textanalyse liegt vor allem darin, daß sehr genaue Vorstellungen darüber existieren, was aus dem Text heraus gelesen werden soll. Daher müssen auch in solch ungünstigen Fällen eines geringen Textkontextes nach Möglichkeit Methoden vermieden werden, denen der Predictor die Kontrolle der Analyse kurzfristig übergibt. Zur Auflösung des Startproblems kann dies nicht völlig vermieden werden, da keine sinnvolle Erwartungshaltung generiert werden kann. Die *antizipierende Diskriminierung* benutzt Methoden, die möglichst schnell und zuverlässig einen Textkontext aufbauen, um dann die Analyse unter der Kontrolle des Predictors fortzusetzen.

Die Anforderungen an solche Methoden sind zum einen eine geringe Dauer, in der dem Predictor die Kontrolle entzogen ist, und zum anderen die Qualität der Antwort hinsichtlich Inhalt und Zuverlässigkeit. Mit dem Begriff der Dauer ist weniger die Programmlaufzeit der Methode gemeint, sondern auch deren Einmaligkeit: die Methode sollte nicht wiederholt angewendet werden müssen. Die inhaltliche Qualität der Antworten sollte auf dem hohem Niveau von Nachrichtentypen oder Schlüssel-Nachrichtenelementen erfolgen. Ist es zu niedrig oder sind die Antworten unzuverlässig, schlägt die antizipierende Diskriminierung dadurch fehl und die iterierende Diskriminierung muß angewendet werden. Genügt die

Methode den Anforderungen, so liefert die Antwort den Hinweis, der einen Start an einem Blatt- oder inneren Knoten rechtfertigt, also einen Hinweis auf bestimmte Nachrichtentypen gibt. Mit einem Blattknoten beginnt eine neue Diastole, d.h. die Instantiierung des Textkontextes und der anschließenden Erweiterung des Erwartungskontextes. Mit einem inneren Knoten muß die Systole noch durch die iterierende Diskriminierung fortgesetzt werden, allerdings mit einem diese abkürzenden Erwartungskontext.

Zum derzeitigen Stand der Implementierung gibt es als verfügbare Methoden die Klassifizierung des Textes nach statistischen Verfahren[7] und nach Mustern[8], deren Ergebnis eine Klassifikation des Briefdokumentes ermöglicht. Die Nachrichtentypen des Nachrichtenmodells werden in einer Liste nach den Wahrscheinlichkeiten geordnet, mit denen sie im Brief vorliegen. Ein eindeutiges Ergebnis wird vom Predictor als Zuordnung zu

einem Nachrichtentyp verstanden, also eines Blattes des Nachrichtenbaums. Liegen die Wahrscheinlichkeiten nahe beieinander, kann der Hinweis bestenfalls als Hinweis auf einen inneren Knoten verstanden werden.

3.2.1.2. Naive Diskriminierung

Gibt es keine Methoden zur antizipierenden Diskriminierung oder liefern diese nur unbrauchbare Ergebnisse, muß der Predictor eine *iterierende Diskriminierung* durchführen, um die Erwartungsmenge auf den Nachrichtentyp, der im Textdokument vorliegt, einzuengen. Da der Erwartungskontext des Predictors, der für die nun beginnende Phase bereitsteht, zu umfangreich ist, um gezielte Vorhersagen zu generieren und der Textkontext leer ist, wird dieser Spezialfall der iterierenden Diskriminierung als *naive Diskriminierung* bezeichnet.

Um einen Kontext zu erarbeiten und sich an den eigenen Haaren aus dem Sumpf der Unwissenheit zu ziehen, benötigt der Predictor geeignete Substantiierer, die von ihrer Leistungsklasse zur Lösung des Startproblems beitragen können. Die naive Anfrage des Predictors stellt als einzige Bedingung die Erwartung eines Elementes auf, mit dem eine gezielte Suche begonnen werden kann. Diese kann als Nachrichtentyp, Nachrichtenelement, CD-Form, Phrase oder Layout- bzw. Logikinformation vorliegen, so daß der Predictor in Abhängigkeit vom Typ der Antwort verschieden reagieren muß. Üblicherweise aber wird sich die Qualität der Antwort höchstens auf dem Niveau einer partiellen CD-Form bewegen.

überleitet. Konnte auch durch Erweiterung der betrachteten Textbereiche keiner der Substantiierer eine Antwort liefern, die eine erfolgreiche Diskriminierung ermöglichte, muß die Analyse abgebrochen werden. Die Möglichkeiten auf verschiedene Sustantiiererantworten zu reagieren werden im folgenden kurz vorgestellt.

Nachrichtentyp

Eine Antwort eines Substantiierers, die auf einen vollständigen Nachrichtentypen hinweist, muß in Abhängigkeit der Fähigkeiten des Substantiierers und der Antwort mit Vorsicht genossen werden. Daher muß der Analysezustand des Predictors, also der aktuelle Kontext, wieder erreichbar sein, falls die Systole oder die Instantiierung fehlschlägt. Wenn die Antwort nur die Klassifizierung, aber keine instantiierten Elemente des Nachrichtentyps enthält, wird sie als Empfehlung eines inneren Knotens gewertet[9]. Wurde sie hingegen von einem Textelement motiviert, z.B. einem Schlüssel des Nachrichtentyps, empfiehlt sie damit einen Blattknoten und aktiviert explizit einen Nachrichtentyp.

Nachrichtenelement

Eine Substantiierer-Antwort in Form eines Nachrichtenelementes, also eine partielle statische oder aktive Konzeptualisierung liefert in der Struktur des Nachrichtenbaums einen Hinweis auf einen Pfad. Da in der Phase der naiven Diskriminierung nur Schlüssel betrachtet werden, ist die Pfadangabe so genau, daß damit ein Nachrichtentyp ausgewählt wird, oder nur eine kleine Anzahl an Alternativen übrig bleibt. Im ersten Fall kann der betreffende Nachrichtentyp aktiviert werden, ansonsten muß die iterierende Diskriminierung die systolische Phase fortsetzen bis eine Auswahl getroffen werden kann. Dazu werden die die alternativen Nachrichtentypen unterscheidenden Nachrichtenelemente zur Generierung von Vorhersagen herangezogen.

CD-Form oder Phrase

Erhält der Predictor als Antwort lediglich ein Textelement (Wort oder Phrase), so muß er dies in eine partielle Konzeptualisierung umwandeln[10]. Dies gelingt nur, wenn der Zugriff auf das Lexikon die CD-Einträge der Phrase oder des Wortes liefert. Mit einer Konzeptualisierung wird eine iterierende Diskriminierung durchgeführt, wenn sie Schlüssel eines Nachrichtentyps sein kann. Dazu muß getestet werden, ob die CD-Form irgendwo im beschnittenen Nachrichtenbaum oder einem Diskriminierungsbaum "paßt". Falls dies der Fall ist, dann wird die gezielte iterierende Diskriminierung gestartet. Anderenfalls kann die gefundene Konzeptualisierung nicht gebraucht werden und wird ignoriert.

---

9 Die Empfehlung des inneren Knotens kann durch die Benutzung eines Fokussierers (siehe [Dittrich 92]) noch präzisiert werden.

10 Es wird davon ausgegangen, daß alle Substantiierer ihre Antworten, die als (partielle) Konzeptualisierung angegeben werden können, auch dergestalt liefern.

Layout-Typ, Layout-Element, Logic-Typ oder Logic-Element

Einem Hinweis auf einen Nachrichtentyp oder ein Nachrichtenelement, der durch die Abbildung einer Layout- oder Logik-Information entsteht, kann nicht unbedingt mit der gleichen Sicherheit vertraut werden, wie einem textbasierten. Der Zustand der Analyse muß daher wieder erreichbar sein. Der Hinweis auf ein Nachrichtenelement durch logische oder Layout-Information wird nur dann berücksichtigt, wenn es sich um einen Hinweis auf einen Schlüssel handelt, dessen Existenz aus dem Text durch eine gezielte iterierende Diskriminierung bewiesen werden muß. Dazu wird die Erwartungshaltung des Predictors auf die Vorhersage dieses Nachrichtenelementes eingeschränkt.

3.2.2 Explizite Aktivierung von Nachrichtentypen

Innerhalb der Steuerungs-Slots `explicit-word-reference`, `explicit-logical-object-reference` und `explicit-layout-reference` der Nachrichtentypen werden Wortbedeutungen, logische Objekte oder Layout-Objekte bezeichnet, die stark mit diesem Typ identifiziert werden. Findet ein Substantiierer z.B. eine Verbphrase, die in `explicit-word-reference` als Schlüssel angegeben wurde, so soll der zugehörige Nachrichtentyp explizit aktiviert werden. Die Idee dieses Verfahrens besteht darin, durch semantisch starke Worte ohne den Umweg der Diskriminierung zum richtigen Nachrichtentyp zu gelangen. Diese Worte haben eine assoziierte Bedeutung, die über die einer einfachen Konzeptualisierung hinausgeht, und direkt einen Nachrichtentypen referenziert. Ist diese Aktivierung falsch, weil das Wort fehlinterpretiert wurde, so muß der Predictor Relationen der impliziten Aktivierung oder Relationen zwischen den Nachrichtentypen ausnutzen.

Die Aktivierung durch explizite Referenz erfolgt also durch Schlüsselphrasen, im Gegensatz zur elementinduzierten Aktivierung, die von der (partiellen) Konzeptualisierung eines Nachrichtenelementes angestoßen wird. Eine derart voreilige Aktivierung kann unangenehme Fehler zur Folge haben. In den folgenden Sätzen

1) Wir bestellen drei Computer.
2) Wir bestellen Ihnen schöne Grüße.
3) Wir bestellen das Feld.

hat das Verb "bestellen" nur einmal die Bedeutung, die als Schlüsselwort zur Aktivierung des Nachrichtentyps `MT-order` erforderlich ist. Die Bedeutung, die "bestellen" in jedem der drei Sätze zugeordnet wird, muß also in Abhängigkeit von anderen Satzteilen, hier vom Objekt des Verbs, abhängig gemacht werden. Wird dies versäumt, so wird der Nachrichtentyp `MT-order` ausgewählt und dem Satz die falsche Konzeptualisierung als `ME-order` zugeordnet. Erfolgt die Aktivierung weniger gezielt, also nicht unmittelbar auf Nachrichtentypen, verringert sich die Zahl der Fehlaktivierungen. Dadurch sind die Anforderungen an die Interpretation gefundener Worte und deren Lexikoneinträge weniger streng.

Kandidaten

Die Kandidaten der Schlüsselphrasen ergeben sich zum Teil aus der Liste der wichtigen Worte eines Nachrichtentyps, wobei die Wichtigkeit durch statistische Methoden bewertet wurde. Die Domäne dient durch Angabe von lexical views in den Constraints des Nachrichtenmodells als weitere Quelle. Allerdings müssen alle Kandidaten, die zur expliziten Aktivierung eines Nachrichtentyps führen sollen, ausdrücklich im `explicit-word-reference`-Slot dieses Nachrichtentyps notiert sein. Zum Beispiel kann die lexical view `LV-customer` als Bedingung der Absenderadresse als `explicit-word-reference` für die Aktivierung des Nachrichtentyps `MT-order` benutzt werden. Es ist keine notwendige Voraussetzung, daß Schlüsselphrasen disjunkt sein müssen. Allerdings sollte ein Schlüssel eine Besonderheit sein, die nur in wenigen Nachrichtentypen auftaucht. Paßt ein Schlüssel auf mehrere Nachrichtentypen, so muß eine Diskriminierung zwischen diesen durchgeführt werden.

Zeitpunkt

Eine explizite Aktivierung erfolgt durch das Auftreten eines semantisch starken Wortes, des Schlüssels. In der späteren Phase der Instantiierung, wenn bereits ein Nachrichtentyp ausgewählt ist, muß das System unterscheiden, ob der Schlüssel zu einer weiteren Aktivierung führt, oder aber vom aktiven Typ konsumiert wird. Durch den Satz 1) wird der Nachrichtentyp `MT-order` explizit aktiviert, falls "bestellen" richtig interpretiert wurde. Der nachfolgende Satz

4) Liefern sie möglichst bald.

darf nicht zur expliziten Aktivierung des Nachrichtentyps `MT-deliver` führen, sondern muß durch `MT-order` als Zeitangabe konsumiert werden. Dieses Beispiel illustriert eine Schwierigkeit mit dem vorliegenden Modell: tatsächlich existiert eine implizite Referenz von `MT-order` zu `MT-deliver`, die für diese Aktivierung zuständig ist. In der eingeschränkten Domäne ist dies in der Regel immer der Fall, da nahezu alle Nachrichtentypen untereinander verbunden sind. Daher würde die explizite Aktivierung hier keinen Fehler darstellen. Um aber ein allgemeineres Modell zu unterstützen, erfolgt die explizite Aktivierung ausschließlich in der Startphase.

Konsequenzen

Durch eine explizite Aktivierung in der Startphase wird die Vorhersagenmenge, also der Erwartungskontext, auf alle noch nicht instantiierten Nachrichtenelemente des aktivierten Nachrichtentyps eingeschränkt. In der Instantiierungsphase werden die Vorhersagen des aktivierten Nachrichtentyps hinzugenommen. Für das Vorgehen des Predictors kann es aber aus Effizienzgründen empfehlenswert sein, nicht alle Vorhersagen sofort, sondern nach ihrer Wichtigkeit oder Reihenfolge geordnet zu aktivieren.

3.2.3 Elementinduzierte Aktivierung von Nachrichtentypen

Die elementinduzierte Auswahl spielt innerhalb der erwartungsgesteuerten Analyse eine wesentliche Rolle. Die Aktivierung durch explizite Referenz auf einen bestimmten Nachrichtentyp bietet die bereits genannten Unsicherheiten, oft werden völlig oder teilweise falsche Aktivierungen erzeugt. Die implizite Aktivierung hingegen basiert nicht auf einer Textinformation sondern tritt nur aufgrund eines aktiven Nachrichtentyps in Kraft. Dieser Kontext ist zwar sehr aussagekräftig, aber er bietet keine Sicherheit für das tatsächliche Auftreten des erwarteten Nachrichtentyps. Der Beweis kann erst durch eine anschließende Diskriminierung erbracht werden. Die elementinduzierte Aktivierung wird durch das Auftreten eines charakteristischen Elementes eines Nachrichtentyps angestoßen. Diese Elemente eines Nachrichtentyps sind durch die Steuerinformation in den Slots `event-induced-reference` (bzw. `layout-induced-reference` und `logical-object-induced-activation`) als solche gekennzeichnet. Sie haben für den Nachrichtentyp, in dem sie auftreten, eine so große Bedeutung, daß dieser aktiviert werden sollte, wenn im Text die mit dem Schlüsselelement verbundene Information gefunden wird.

Die elementinduzierte Aktivierung kann, abhängig von der Domäne, den häufigsten Fall der Aktivierungsarten stellen: In einer Domäne, die eine breite Streuung der zu erwartenden Texte hat und in der es viele verschiedene Wege gibt, über einen Sachverhalt zu sprechen, lassen sich wenige Schlüsselphrasen angeben. Für eine eingeschränkte und dazu noch formalisierte Domäne können sichere Schlüsselphrasen angegeben werden und so den Schwerpunkt zugunsten der expliziten Aktivierung verschieben. Innerhalb des Nachrichtenmodells der Geschäftsbriefe ließe sich diese Verschiebung durch eine strenge Festlegung der Ausprägung der Nachrichtentypen erreichen, womit akzeptierbare Geschäftsbriefe dann eher Formulare wären.

In der Startphase kann durch Instantiierung eines als wichtig markierten Nachrichtenelementes eine Aktivierung des entsprechenden Nachrichtentyps erfolgen. Die Aktivierung basiert also auf dem Auftauchen eines Konzeptes im Text, statt lediglich einer Schlüsselphrase, wie im Fall der expliziten Aktivierung. Solange noch kein Nachrichtentyp favorisiert ist, werden die Einträge der `element-induced`-Slots als einzige Erwartungen des Predictors aufgestellt. Dies kann am effizientesten durch die Diskriminierungsbäume geschehen, alternativ kann jedoch jede andere Inkarnation des Nachrichtenmodells benutzt werden.

Kandidaten

Die Kandidaten der elementinduzierten Aktivierung werden durch die Einträge der `element-induced`-Slots der Nachrichtentypen und der damit assoziierten Nachrichtenelemente bereitgestellt. Der Predictor beachtet lediglich die so gekennzeichneten Elemente für die elementinduzierte Aktivierung. Jede Vorhersage, also jedes Nachrichtenelement in der Struktur des Nachrichtenmodells könnte als Kandidat für die elementinduzierte Aktivierung dienen. Allerdings würde damit das Verfahren der

Diskriminierung erschwert, weil zuviele ähnlich strukturierte Alternativen zu betrachten sind. Die Diskriminierung würde ihren Sinn, nämlich schnell zu einem Nachrichtenelement zu führen, verlieren. Daher werden nur ausgezeichnete Elemente eines Nachrichtentypen für die elementinduzierte Aktivierung betrachtet, diese sind in den Slots zur Element- bzw Objekt- und Layout-induzierten Aktivierung notiert.

Zeitpunkt

Die elementinduzierte Aktivierung findet sowohl in der Startphase, als auch in der Phase der Instantiierung, also wenn bereits ein Nachrichtentyp aktiviert ist, statt. Ist nach der Startphase ein Nachrichtentyp aktiviert, so werden durch die Verfahren der impliziten Aktivierung die Elemente, die zur Aktivierung implizit verbundener Nachrichtentypen führen können, in die Vorhersagenmenge aufgenommen. Dieses Vorgehen wird genau in Kapitel 3.5.4 erklärt.

3.3 Die Diskriminierungsphase

Der Erwartungskontext, der dem Predictor zur Auswahl steht, pulsiert im Laufe der Analyse, ausgehend vom Maximalwert aller Nachrichtenelemente aller Nachrichtentypen, zwischen dem Minimum eines Nachrichtenelementes und einem mittleren Wert mit mehreren Nachrichtenelementen. In der systolischen oder Diskriminierungsphase wird der Erwartungskontextes auf einen Nachrichtentyp eingeschränkt, falls dies noch nicht durch die Startphase erreicht wurde. Sukzessiv werden in einer Schleife (*iterierende Diskriminierung*) dann die Vorhersagen seiner Nachrichtenelemente generiert und als Erwartungen an die Substantiierer übergeben. Das Ergebnis wird in der diastolischen Phase als Instanz im Textkontext der Analyse eingetragen. Nach wiederholten Wechsel zwischen Systole und Diastole endet die Analyse erfolgreich - d.h. alle Erwartungen wurden befriedigt - oder durch einen unbehebbaren Fehler.

In der Diskriminierungsphase wird das Verfahren der iterierenden Diskriminierung angewendet. Dies stellt dann, wenn die Startphase nicht durch eine Diskriminierung mit Diskriminierungsbäumen vorgenommen wurde, die Fortsetzung dieses Verfahrens auf einem spezielleren Erwartungskontext dar. Es beinhaltet die Berechnung einer Diskriminante und die Generierung der Substantiiereranfragen. Die Bearbeitung der Antworten der Substantiierer wird von der diastolischen Phase übernommen. Nach der diastolischen Phase, also dem Eintragen der Antworten und der Anpassung der Erwartungen aufgrund dieses neuen Textkontextes, dient die Diskriminierung wiederum zur Einschränkung und Auswahl der im Dokument vorliegenden Nachrichtenelemente und der durch implizite Verkettung verbundenen Nachrichtentypen.

Iterierende Diskriminierung

Die Phase der naiven Diskriminierung versucht aus einem zu umfangreichem Erwartungskontext einen Status zu erreichen, indem der Textkontext nicht mehr leer ist und

die Menge der Erwartungen präzisiert werden konnte. Sie führt damit in den meisten Fällen zur *iterierenden Diskriminierung*, die nicht nur die Schlüsselelemente der Nachrichtentypen, sondern innerhalb der systolisch-diastolischen Schleife auch die restlichen Elemente diskriminiert.

Der Erwartungskontext der iterierenden Diskriminierung ist nicht auf einen Nachrichtentyp beschränkt, wenn die naive Diskriminierung nicht so erfolgreich war, oder in der Diastole implizite Aktivierungen gestartet wurden, d.h. weitere Nachrichtentypen werden im vorliegenden Dokument vermutet. Im ersten Fall muß aus den noch in Frage kommenden Nachrichtenelementen verschiedener Nachrichtentypen eine Diskriminante als Vorhersage berechnet werden, deren Überprüfung durch einen Substantiierer die Diskriminierung auf einen ermöglicht. Die Art der Substantiierer-Antwort auf die Erwartung der Diskriminante entscheidet dann, welches Nachrichtenelement gewonnen hat. Unter Umständen ist mit einer Diskriminante noch keine eindeutige Entscheidung herbeizuführen, so daß dieses Verfahren wiederholt angewendet werden muß.

3.4 Die Instantiierungsphase

Ist die Menge der Erwartungen bereits auf ein Nachrichtenelement eingeschränkt, so muß zu dessen Instantiierung die Erwartungshaltung auf alle Slots dieses Nachrichtenelementes erweitert werden. Diese werden dann durch Substantiiereranfragen in der im Nachrichtenmodell vorgegebenen Reihenfolge instantiiert. Dabei bestehen verschiedene Möglichkeiten der Qualität der Antworten (siehe 3.5.3): die Erwartung konnte nicht bestätigt werden, die Erwartung wurde bestätigt oder die Erwartung wurde übertroffen. Im ersten Fall kann ein Fehler anfallen, wenn die Erwartung eine besonders wichtige ist, die entscheidend für das Vorliegen des erwarteten Nachrichtentyps ist. Eine bestätigte Erwartung wird im Textkontext eingetragen. Das Phänomen einer übertroffenen Erwartung kann dadurch entstehen, daß ein Substantiierer zusätzlich zur gewünschten Antwort noch den zugehörigen Kontext liefert. Wird z.B. ein Aktionswort (lexical view `lv-action`, also ein Verb) erwartet, kann von einem Substantiierer, der die syntaktische Struktur Agent–Aktion–Objekt erkennen kann, diese instantiiert zurückgegeben werden. Die Notwendigkeit weiterer Substantiiereranfragen wird dadurch vermieden.

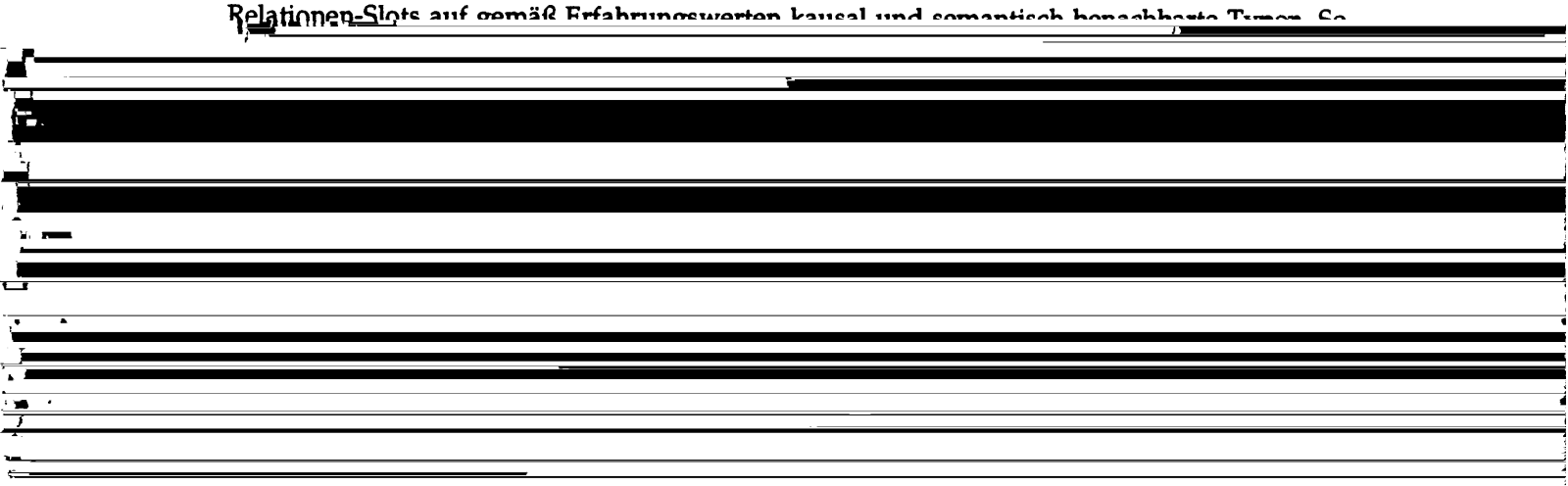
Die Aufgaben des Predictors in der Instantiierungsphase bestehen darin, die Erwartungsmenge auf alle Nachrichtenelemente des aktiven Nachrichtentyps zu erweitern (Kapitel 3.5.1). Die Menge der Vorhersagen wird in der durch die Steuerinformation im Nachrichtentypen vorgegebenen Reihenfolge als Anfragen an die Substantiierer (Kapitel 3.5.2) weitergegeben und vom Predictor interpretiert (Kapitel 3.5.3). Die Interpretation beinhaltet das einfache Eintragen der Substantiiererantwort oder die Anwendung von Regeln. Wie in jeder der vorangegangenen Phasen muß auch das Fehlschlagen der Diastole behandelt werden.

3.4.1 Erweiterung der Erwartungsmenge

Der Erwartungskontext besteht nach der Systole aus einem aktivierten Nachrichtentyp. Um nun alle Elemente dieses Typs aus dem Text instantiieren zu können, muß zuerst die Erwartungshaltung auf alle Nachrichtenelemente dieses Typs ausgedehnt werden, die dann in der durch die Nachrichtentypen vorgegebenen Wichtigkeit und Reihenfolge als Anfragen an die Substantiierer weitergegeben werden. Innerhalb der Sicht der Nachrichtentypen als Baumstruktur bedeutet dies, daß alle Elemente entlang des Pfades in die Vorhersagenmenge aufgenommen werden. Die Verweise zur impliziten Aktivierung bewirken, daß ein entsprechender Multi-Nachrichtentyp gesucht oder neu erzeugt wird. Damit treten die impliziten Aktivierungen in Kraft, d.h. deren Schlüssel erweitern ebenfalls die Erwartungsmenge.

3.4.1.1. Implizite Aktivierung von Nachrichtentypen

Nachrichtentypen, die auf welche Art auch immer aktiviert wurden, verweisen durch Relationen-Slots auf gemäß Erfahrungswerten kausal und semantisch benachbarte Typen. So

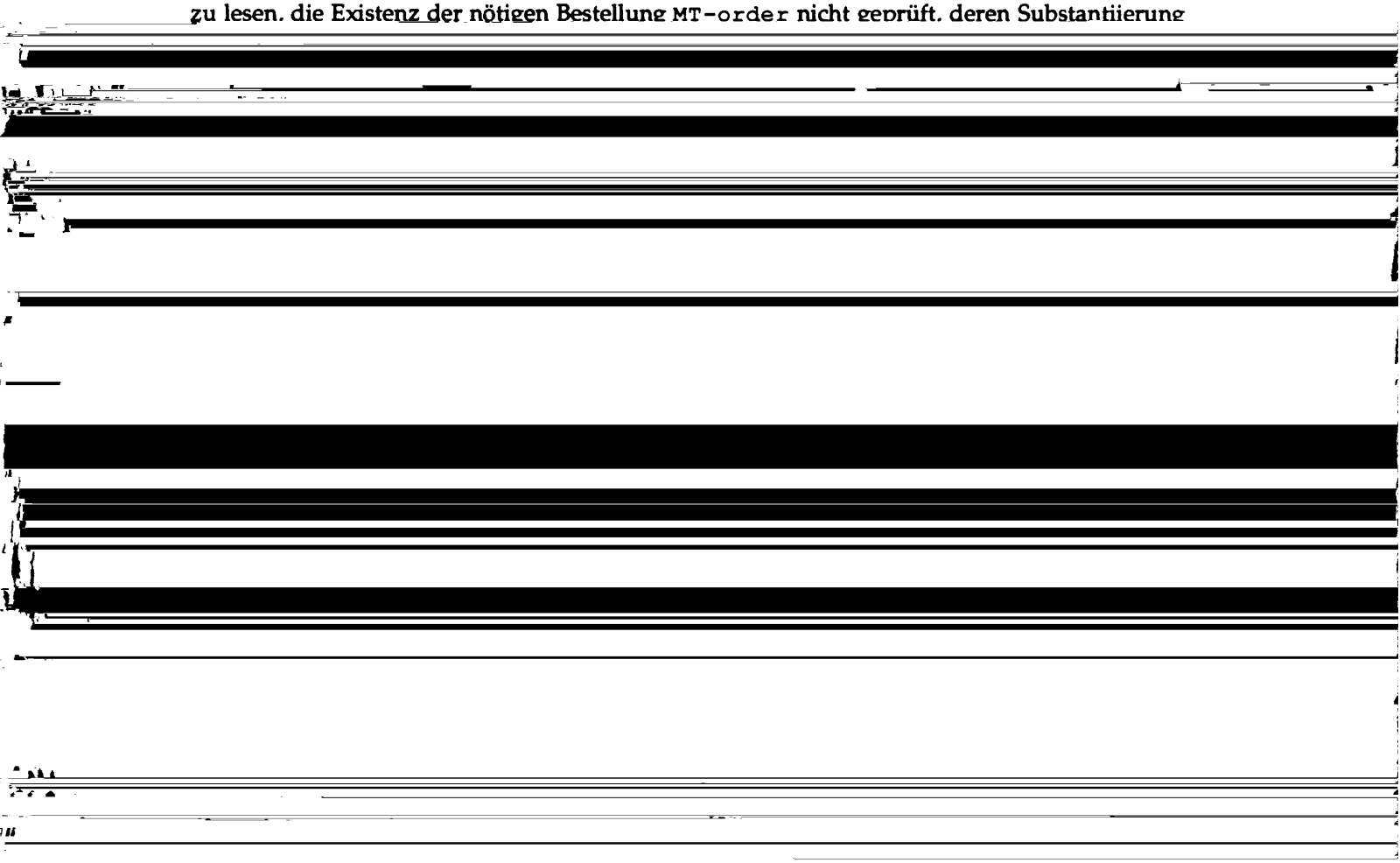


enthält der Nachrichtentyp `MT-order` (Bestellung) Verweise auf `MT-orderchange` (Bestelländerung) und wie dieser auf `MT-acknowledge` (Bestellbestätigung). Der Eintrag zur impliziten Aktivierung eines Nachrichtentyps ist eine durch die implizite Referenz gerichtete Kante des Multi-Nachrichtentyps; der Verweis auf einen Nachrichtentyp ist auch ein Verweis auf den zugehörigen Multi-Nachrichtentyp. In der Phase der Instantiierung müssen die Einträge zur impliziten Aktivierung in den Steuerungseinheiten `implicit-reference` betrachtet werden.

Bedeutung der impliziten Aktivierung

Der Eintrag zur impliziten Aktivierung besagt, daß die Möglichkeit besteht, daß dem aktuellen Nachrichtentyp verwandte Nachrichtentypen ebenfalls im Text erwähnt werden (in einer Rechnung wird auf die Bestellung Bezug genommen), oder aber umgekehrt ein Erwartungskontext besteht, in den dieser Nachrichtentyp eingebettet werden kann (zur vorliegenden Rechnung wurde bereits der Bestellbrief analysiert und der entsprechende Nachrichtentyp `mt-order` und Multi-Nachrichtentyp erzeugt). Das bedeutet, daß im Erwartungskontext nach einem Multi-Nachrichtentyp gesucht wird, der den gerade aktiven Nachrichtentyp aufnehmen kann, oder aber die Instanz eines passenden Multi-

Elemente des Graphen unberücksichtigt bleiben, die in der Kausalkette vorher liegen. Die Interpretation der impliziten Verkettung als kausale Abfolge von Nachrichtentypen in einem Dokument oder einer mehrere Briefe umfassenden Korrespondenz ist diskutierbar: Im Beispiel des Nachrichtentyps `MT-orderchange` wird, ohne die Aktivierungsinformation rückwärts zu lesen, die Existenz der nötigen Bestellung `MT-order` nicht geprüft, deren Substantiierung



loren, wenn diese später analysiert wird. Das Ergebnis einer Bestelländerung ohne die zugehörige Bestellung kann die auf die Analyse folgende Interpretation nicht sinnvoll einordnen. Die Argumentation, die für das gerichtete Lesen der impliziten Verkettungen spricht, geht berechtigt davon aus, daß die Eingangsreihenfolge der Briefe nicht willkürlich erfolgen soll. Verbindungen, die in beiden Richtungen gelten sollen, müssen also in beiden Nachrichtentypen und in den zugehörigen Multi-Nachrichtentypen notiert werden, so daß durch die Definition des Nachrichtentyps die Beziehung klar wird, anstatt von der Interpretation des Predictors abzuhängen. Der Nachrichtentyp `MT-orderchange` muß daher den impliziten Verweis auf `MT-order` enthalten.

Aktivierung nach der Instantiierungs-Phase

Solange der Predictor erfolgreiche Antworten von den Substantiierern für die Instantiierung des ausgewählten Nachrichtentyps erhält, bleibt die Aktivierung aus. Die Vorhersagen der Schlüssel der implizit verbundenen Nachrichtentypen werden nicht in die Vorhersagenmenge aufgenommen. Können die Erwartungen des Predictors nicht mehr befriedigt werden, müssen die Schlüsselvorhersagen der implizit verbundenen Nachrichtentypen aktiviert werden. Der Nachteil dieser Strategie besteht darin, daß während der Instantiierung überlesener Text für die implizit aktivierten Nachrichtentypen von Bedeutung gewesen sein könnte. Überlesener Text muß also als solcher markiert werden, um später zur Überprüfung der impliziten Aktivierungen untersucht zu werden. Wenn die Menge der neuen Erwartungen impliziter

Kandidaten

Durch die genaue Untersuchung einer Vielzahl an Briefen der Domäne kann das Verbindungsnetz der Nachrichtentypen aufgestellt und als implizite Aktivierungsinformation und Multi-Nachrichtentyp im Nachrichtenmodell kodiert werden. Innerhalb der eingeschränkten Domäne der Geschäftsbriefe existieren zwischen nahezu allen Nachrichtentypen implizite Beziehungen, d.h. daß mehrere Multi-Nachrichtentypen existieren, die sehr viele Nachrichtentypen enthalten. Mit der impliziten Aktivierung werden jedoch nur die unmittelbar durch einen Eintrag im `implicit-reference`-Slot verknüpften Nachrichtentypen aktiviert, nicht aber alle Nachrichtentypen des zugehörigen Multi-Nachrichtentyps.

Konsequenzen

Durch eine implizite Aktivierung wird ein Hinweis auf mögliche andere Nachrichtentypen gegeben, die im aktuellen Kontext auftauchen und mit dem aktiven Typ einen Multi-Nachrichtentyp bilden könnten. Gibt es im aktuellen Kontext des Systems bereits einen Multi-Nachrichtentyp dieser Art, so kann der gefundene Nachrichtentyp diesem zugeordnet werden, sofern die Constraints des Multi-Nachrichtentyps erfüllt sind. Ist eine der beiden Bedingungen nicht erfüllt, muß ein neuer Multi-Nachrichtentyp erzeugt werden.

Der Erwartungskontext wird nun um die Schlüssel, also die Einträge in den Slots `explicit-reference` und `element-induced-activation` dieser Nachrichtentypen erweitert. Erst wenn ein Schlüssel im Text gefunden wurde, werden die restlichen Elemente des entsprechenden Nachrichtentyps aktiv. Kann ein Schlüssel nicht gefunden werden, d.h. die durch die implizite Aktivierungsinformation gegebene Möglichkeit des auftauchens eines weiteren Nachrichtentyps liegt im Dokument nicht vor, so werden die Erwartungen der Schlüssel aus dem Erwartungskontext gelöscht. Ihr Fehlen bedeutet hier keinen Fehler, gleichwohl sie als Aktivierungsinformationen eine hohe Wichtigkeit haben. Solange der Nachrichtentyp nicht aktiviert ist, bleibt diese unberücksichtigt.

Das Auftreten eines Schlüssels im Text und die Instantiierung führt zur elementinduzierten Aktivierung des implizit verbundenen Nachrichtentyps,

3.4.1.2. Elementinduzierte Aktivierung von Nachrichtentypen in der Instantiierungsphase

In der Instantiierungsphase kann, ähnlich wie in der Startphase, eine elementinduzierte Aktivierung erfolgen. Hier findet die Aktivierung jedoch entweder in einem Kontext statt, der mindestens einen aktiven Nachrichtentyp enthält, also die Vorgabe einer impliziten Aktivierung, oder durch die sukzessiv Instantiierung des Schlüsselelementes in der Schleife zwischen Diskriminierung und Instantiierung.

3.4.2 Instantiierung durch Substantiiereranfragen

Der Erwartungskontext des Predictors besteht aus allen Elementen des aktiven Nachrichtentyps und den Schlüsseln der implizit aktivierten Typen. Diese Erwartungsmenge stellt ein Fragment des Nachrichtenbaumes dar. Für den aktiven Typ werden die Erwartungen in der durch das Nachrichtenmodell vorgegebenen Reihenfolge als Substantiiereranfragen produziert, die impliziten Aktivierungen sind als isolierte Knoten im Baum latente Elemente der Erwartungsmenge. Auf diesem Baumfragment wird eine iterierende Diskriminierung und Instantiierung durchgeführt, bis alle Erwartungen erfüllt wurden oder ein Fehler die Entscheidung für den Nachrichtentyp revidieren läßt.

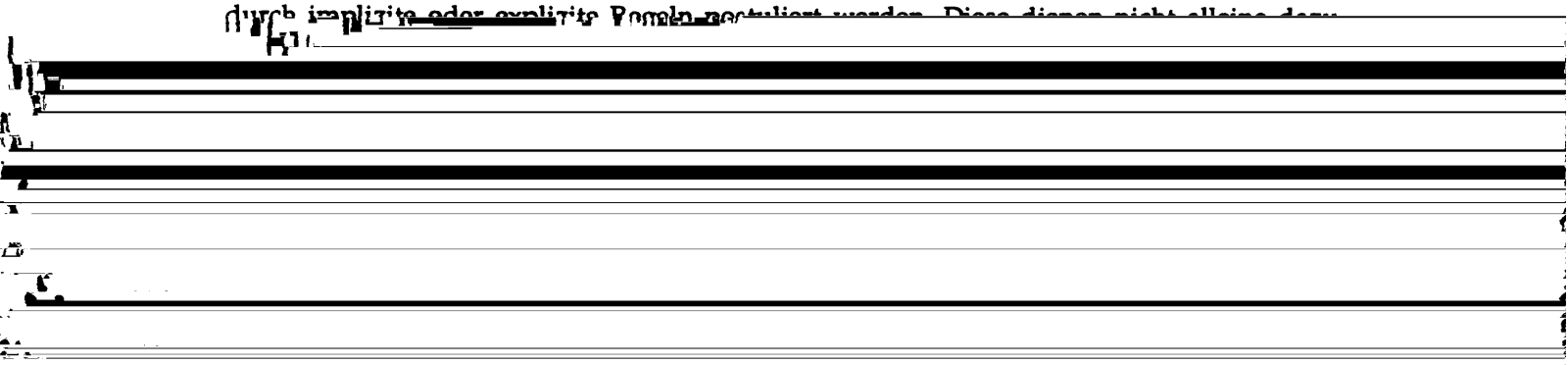
Vor der Instantiierungsschleife werden aus den Erwartungen und den zugehörigen Steuerinformationen die Substantiiereranfragen erzeugt. Dazu wird in einer erneuten Phase der iterierenden Diskriminierung eine Diskriminante berechnet, bis ein Nachrichtenelement isoliert ist. Die Diskriminante, und schließlich auch die übrigen Slots eines Nachrichtenelementes, werden als Substantiierer-Anfragen generiert.

Der für die Erwartungen nötige Substantiierer wird aus den `need-substantiator`-Slots

erste Eintrag jedes Slots die beste Empfehlung ist. Wenn die Erwartungen widersprüchliche Substantiierer benötigen, kann entweder der allgemeinste Substantiierer benutzt werden oder aber seriell oder parallel alle empfohlenen Substantiierer. Die Vorgaben der `expected-location`-Slots können dann gezielt benutzt werden, wenn die Diskriminante sukzessiv als Erwartung an die Substantiierer weitergegeben wird. In dieser Instantiierungs-Schleife kann es erforderlich sein, die Wahl des benutzten Substantiierers zu ändern, die Anfrage wird also erneut an einen anderen Substantiierer gestellt. Schlägt eine Anforderung fehl, kann das für die Analyse den völligen Mißerfolg, die Notwendigkeit zum *Backtracking* (siehe 3.6) oder die Anwendung von Regeln bedeuten.

zwischen dem Minimum einer Zeichenkette, dem Mittelwert einer Konzeptualisierung und den Maxima von Nachrichtenelementen oder, in Verbindung mit Multi-Nachrichtentypen, von Nachrichtentypen liegen. Wenn die Anfrage keine Diskriminante war, sondern gezielt eine Informationseinheit erwartet, kann die gefundene Information sofort in dem entsprechenden Nachrichtenelement eingetragen werden. Ansonsten muß sie in einer Hilfs-CD-Form zwischengespeichert werden. Sobald die Diskriminierung mit der Auswahl eines Nachrichtenelementes abgeschlossen ist, werden die Inhalte der Hilfs-CD-Form in die entsprechenden Slots des Elementes kopiert.

Das Eintragen eines Elementes kann aber noch weiterreichende Konsequenzen haben, die durch implizite oder explizite Regeln postuliert werden. Diese dienen nicht alleine dazu



nicht aus dem Text zu erfüllende Erwartungen aus anderen Werten des Textkontextes zu belegen, sondern auch um Beziehungen zwischen diesen aufzustellen. Es gibt also Regeln, die optional angewendet werden können (`rule-can`), wenn eine Substantiierung fehlschlug, obligatorische Regeln (`rule-must`), die eine zwingende Beziehung aufstellen und stets getestet werden müssen und die Regeln für Standardwerte (`rule-default`). Dabei wird der `importance`-Eintrag eines Elementes nicht von der Art der Regel berührt, d.h. daß auch wichtige Elemente optionale Regeln haben können, nicht aber unwichtige Elemente obligatorische.

Ein Fehler der Analyse liegt vor, wenn eine obligatorische Regel oder eine optionale Regel für ein wichtiges Element fehlschlägt. Davon sind die wichtigen Elemente, die als Schlüssel für eine implizite Aktivierung dienen sollen, zunächst nicht berührt. Ihr Fehlen bedeutet keinen Fehler, sondern lediglich, daß die durch die implizite Referenz aufgestellte Option im vorliegenden Dokument nicht wahrgenommen wurde, eine Aktivierungsinformation also nicht zur Aktivierung geführt hat.

3.4.3.2. Regelanwendung

In diesem Unterkapitel werden die Aktionen des Predictors beschrieben, die auf die Antworten eines oder mehrerer Substantiierer aufgrund der Anwendung von Regeln folgen können. Die Regeln zur Festlegung der Aktionen des Predictors sind im Nachrichtenmodell zwischen Informationseinheiten der CD-Formen, der Nachrichtenelemente und der Nachrichtentypen definiert. Hierbei wird zwischen den Regelarten `rule-must`, `rule-can` und `rule-default` unterschieden. Erstere haben die höchste Priorität und müssen angewendet werden und ihr Fehlschlagen signalisiert ein Scheitern der Analyse. Die zweite Klasse kann angewendet werden, um z.B. nicht durch Substantiiererantworten gefüllte Slots mit Werten anderer Slots zu belegen. Durch Defaultregeln können den Slots Standardbelegungen zugewiesen werden. Der Predictor betrachtet während der Analyse einer Informationseinheit die auf gleicher Ebene definierten Slots `rule-must`, `rule-can` und `rule-default`. Die auf

tigt[11]. Die Regeln unterscheiden sich neben der Art nach dem Ort ihrer Definition und den Rollenfüllern, die inferiert werden können: innerhalb eines Nachrichtenelementes zwischen Slots, innerhalb eines Nachrichtentyps zwischen Slots und Nachrichtenelementen oder aber in einem Multi-Nachrichtentyp zwischen Slots, Nachrichtenelementen und Nachrichtentypen.

Die Anwendbarkeit von Regeln wird in Abhängigkeit vom Erwartungskontext bewertet: befindet sich der Predictor in der systolischen Phase ohne daß schon ein Nachrichtentyp aktiv ist, macht eine Regelanwendung kaum Sinn, zumal die zur Inferenz benötigte Information anderer Slots noch nicht bereitsteht. Innerhalb dieser Phase werden daher keine Regeln angewendet. Außerdem würde die frühzeitige Regelanwendung verhindern, daß Slots durch Substantiiereranfragen gefüllt werden.

Ist die Diskriminierung soweit fortgeschritten, daß ein Nachrichtentyp aktiviert ist, dann können dessen Regeln und die seiner Nachrichtenelemente und Slots angewendet werden. Regeln, die in dem zugehörigen Multi-Nachrichtentyp aufgestellt sind, werden dann angewendet, wenn das Einbinden eines neuen Nachrichtentyps dies erfordert.

Regeln zwischen Slots

Die Regelanwendung zur Herleitung von Informationseinheiten innerhalb eines Nachrichtenelementes wird solange verzögert, bis der Erwartungskontext sicher ist, d.h. ein Nachrichtentyp ausgewählt ist und die Instantiierungsversuche der Substantiierer fehlgeschlagen sind.

Das gleiche Vorgehen wird für die Auswertung einer Regelbeziehung zwischen Slots, die in einem Nachrichtentyp oder einem Multi-Nachrichtentyp aufgestellt wurde, verwendet. Die Regelauswertung wird solange verzögert, bis beide Nachrichtenelemente partiell substantiiert wurden. Die Aktion, die eine Regel bedingt, besteht gewöhnlich aus dem Kopieren der Inhalte eines Slots in die eines anderen.

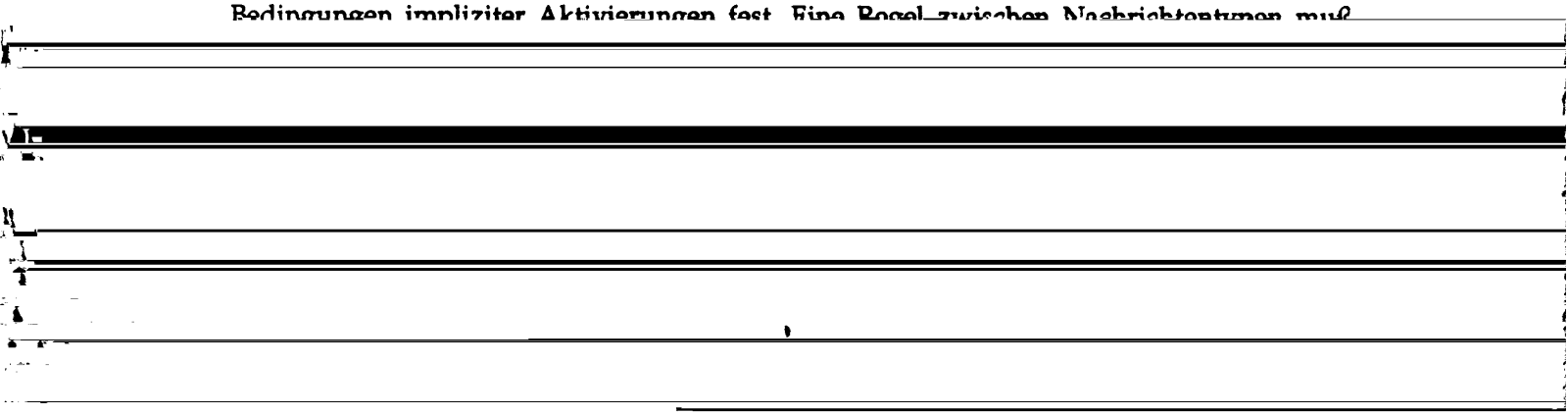
Regeln zwischen Nachrichtenelementen

Ein Beispiel einer Regelbeziehung, die zwischen Nachrichtenelementen aufgestellt wird, ist die Bedingung, daß in einem Bestellbrief der Besteller mit dem Absender identisch sein muß. Wenn der Versuch fehlgeschlagen ist, den Besteller im Text zu substantiieren, kann die Regel angewendet und der Inhalt kopiert werden. Dabei sind allerdings einige Hürden zu überwin-

Nachrichtentypen und nicht als Bestandteil des Predictor definiert werden. Im folgenden wird von der Existenz dieser Funktion ausgegangen, wenn vom Kopieren von Elementen oder Slots die Rede ist.

Regeln zwischen Nachrichtentypen

Dieser Typus, der nur innerhalb von Multi-Nachrichtentypen auftritt, legt die Beziehungen zwischen Elementen der beteiligten Nachrichtentypen und als obligatorische Regeln auch die Bedingungen impliziter Aktivierungen fest. Eine Regel zwischen Nachrichtentypen muß



überprüft werden, bevor die implizite Aktivierung mit der Erzeugung eines neuen Multi-Nachrichtentyps oder dem Einbinden in einen bereits bestehenden abgeschlossen wird. Damit wird erreicht, daß für eine neue Bestellung auch ein neuer Multi-Nachrichtentyp erzeugt wird. Zum Beispiel können zwischen den Nachrichtentypen im Multi-Nachrichtentyp `MMT-order-process`

```
(defclass MMT-order-process ()
   (   (advertising :type MT-advertising)
       (offering :type MT-offer)
       (ordering :type MT-order)
       (changingorder :type MT-change-order)
       (delivering :type MT-deliver)
       (calculating :type MT-calculation))  )
```

folgende zwingenden Beziehungen durch `rule-must`-Regeln gelten:

- die Agenten in allen Nachrichtentypen sind die gleichen
- die zeitliche Aufeinanderfolge der Nachrichtentypen ist Werbung, Angebot, Bestellung, Bestelländerung, Lieferung und Rechnung.
- das Objekt der Bestellung ist auch das der Lieferung
- usw.usf.

3.4.3.2.1 Standard Regeln

Eine Standardregel `rule-default` wird angewendet, wenn ein Slot nicht durch eine Substantiiererantwort gefüllt wurde und keine andere Regel einen Füller liefert. Das Ergebnis der Regelanwendung ist das Eintragen des vorgegebenen Wertes. Der Predictor interpretiert die im folgenden Beispiel angegebene Regel durch Ersetzen von `*self*` mit der Instanz des betroffenen Slots (hier `direction-from`) und Evaluierung durch die Lispfunktion `eval`.

```
(rule-default
```

Die Regeln der Klasse `rule-default` sorgen dafür, daß alle Slots einer Erwartung gefüllt werden. So gibt es im Nachrichtenelement `me-order` auch die Slots `dir-from` und `dir-to`, die aber nur selten aus dem Satzkontext der Bestellung (z.B. in einer tabellarischen Auflistung) gefüllt werden können. Stattdessen können sie aus den Adressen von Sender und Empfänger des Briefes gewonnen werden[12].

3.4.3.2.2 Optionale Regeln

Eine optionale Regelanwendung wird dann nötig, wenn ein Substantiierer nicht in der Lage ist, die erwarteten Informationen im Text zu finden. Im Gegensatz zu den Defaultregeln werden optionale Regeln für Slots definiert, für die nicht sicher ist, daß das Füllen einer Informationseinheit immer durch Kopieren einer anderen erreicht werden kann. Eine Antwort dieses *"Inferenz-Substantiierers"* (der ein Teil des Predictors ist) ist daher immer mit einer Angabe des Zweifels versehen, die ebenfalls im `phrase-found`-Aspekt notiert wird. Sie ist stets größer als der Standardwert 1 einer aus dem Text substantiierten Antwort. Im Anschluß an die Analyse kann aus den Zweifeln aller Informationseinheiten der Zweifelswert des gesamten Nachrichtentyps berechnet werden.

3.4.3.2.3 Obligatorische Regeln

Die obligatorischen Regeln `rule-must` stellen Bedingungen auf, die zwischen wichtigen Elementen eines Multi-Nachrichtentyps, eines Nachrichtentyps oder eines Nachrichtenelementes gelten müssen. Dies können, wie bei den übrigen Regeln auch, Slot-, Nachrichtenelement- oder Nachrichtentyp-Beziehungen sein. Die Auswahl der anwendbaren Regeln erfolgt auf die gleiche Weise wie die der optionalen, von der innersten zur äußersten Struktur. Der Eintrag durch eine obligatorische Regel wird wie der von optionalen Regeln mit einer Angabe des Zweifels versehen, die im `phrase-found`-Aspekt notiert wird. Sie ist stets größer als der Standardwert "1" einer aus dem Text substantiierten Antwort.

3.4.3.3. Auflösung von Alternativen

Da die instantiierten Erwartungen als Ergebnis einer Substantiierung aufgrund einer fehlerbehafteten Texterkennung oft mehrdeutig sind, ist es nötig die Alternativen nach ihrer Wahrscheinlichkeit zu ordnen, um so eine bevorzugen und die falschen ausschliessen zu können.

---

[12] Dabei gehen wir vereinfachend davon aus, daß eine Bestellung nicht über einen Vermittler abläuft.

Wie aber ist das möglich? Die einzige zur Verfügung stehende Information liefern die semantischen Constraints. Diese aber haben jede der Alternativen zugelassen. Hier können lediglich Regeln, die ebenfalls eine Belegung einer Erwartung ermöglichen, durch Abgleich ihres Ergebnisses mit denen der Substantiierung helfen. Dazu werden mit mehrdeutigen Substantiiererantworten belegte Informationseinheiten die mit Regeln verbunden sind, auf ihre Konsistenz überprüft. Die Hoffnung besteht darin, daß der Schnitt der beiden Belegungen eine Auflösung der Mehrdeutigkeit möglich macht.

3.5 Fehlerbehandlung

Die Behandlung von Fehlern ist in allen Phasen der Diskriminierung und Instantiierung nötig, da die *Substantiierer*-Anfragen oft nicht das Ergebnis liefern können, das von ihnen erwartet wird. Fehler können zum Abbruch der Analyse führen, womit das eingelesene Dokument nicht mehr einem Nachrichtentyp zugeordnet werden kann. Durch das Fehlen wichtiger Elemente und die Unmöglichkeit der Diskriminierung bleibt der Erwartungskontext so umfangreich, daß das System keine Erwartungen generieren kann. Das Ziel der erwartungsgesteuerten Textanalyse soll aber die Vermeidung von Fehlern, also eine hohe Robustheit sein. Auch wenn abgebrochen werden muß, soll ein Maß an Information aus dem Text extrahiert worden sein.

In der Sichtweise der Baumstrukturierung des Nachrichtenmodells kann auch aus einer gescheiterten Analyse noch einiges herausgelesen werden, so daß einige Informationen des Dokumentes erkannt werden können. Der Maßstab dafür liegt in der Menge der aufgebauten Konzeptualisierungen (Textkontext), die vor Auftreten des Fehlers erzeugt wurden und der Position der Analyse im Nachrichtenbaum, also des Erwartungskontextes. Einem Analyseergebnis, das an einem inneren Knoten scheitert, kann innerhalb der Grenzen, die durch die nachfolgenden Blätter aufgestellt werden, eine ungefähre Bedeutung im Sinne eines *best fit* zugewiesen werden.

Fehlerursachen entstehen dadurch, daß der Predictor auf eine Anfrage keine, eine falsche oder eine mehrdeutige Antwort erhält. Die Einschätzung von Fehlern und die Reaktion darauf hängt davon ab, auf welchem Niveau der Predictor keine Antwort bekommt, und wie wichtig die Antwort wäre. In Kapitel 3.6.1 werden die Fehler, die während der Instantiierung vorkommen können, aufgezählt. Sie bedingen Fehler in der Diskriminierung, die abschließend in 3.6.2 behandelt werden.

3.5.1 Instantiierungsfehler

Unter dem Begriff der Instantiierungsfehler werden die Reaktionen des Predictors angegeben, wenn eine Erwartung eines Slots, Nachrichtenelementes oder Nachrichtentyps nicht erfüllt werden konnte.

Instantiierung von Slots

Wenn ein Fehler in der Slot-Instantiierung auftritt, also eine Anfrage an einen Substantiierer nicht beantwortet werden konnte, werden zunächst die alternativen Beschaffungsmethoden probiert: In einer Schleife werden die in dem `need-substantiator`-Aspekt angegebenen Substantiierer auf den durch den `expected-locations`-Aspekt empfohlenen Positionen des Dokumentes angestoßen. Bleibt dieser Versuch ohne Ergebnis, müssen die Regeln des Slots angewendet werden. Schlagen diese fehl oder es sind keine Regeln vorhanden, bedeutet dies ein endgültiges Scheitern des Instantiierungsversuches.

Der Predictor bewertet das Fehlen eines Slots nach der Wichtigkeit, die im `importance`-Eintrag angegeben wurde und der Möglichkeit einer späteren Regelanwendung durch einen Zweifels-Wert. Fehlt ein sehr wichtiges Element (`importance 1`) und es gibt keine Regel, wie z.B. für das Objekt einer Aktion, dann kann das zugehörige Nachrichtenelement nicht als substantiiert betrachtet werden, der Slot wird mit dem Zweifelswert "∞" bewertet. War der Slot wichtig (`importance 2`), toleriert der Predictor das Nachrichtenelement mit dem Zweifelswert "2", war er unwichtig (`importance 3`) mit "1". Liegen für den Slot Regeln vor, die aber nur verzögert ausgewertet werden können, so wird der Slot mit dem Zweifelswert von "2" akzeptiert, der mit dem Zweifel der Regelanwendung multipliziert wird.

Instantiierung von Nachrichtenelementen

Ein Fehler in der Instantiierung eines Nachrichtenelementes liegt vor, wenn das Vertrauensmaß $V_{\texttt{Nachrichtenelement}}$ zu gering ist. Das Vertrauensmaß eines Nachrichtenelementes wird durch die folgende Formel überprüft:

$$V_{\texttt{Nachrichtenelement}} = \frac{\sum_{\texttt{Slot}} (\texttt{Max}_{\texttt{importance}} - \texttt{importance}_{\texttt{Slot}})}{\sum_{\texttt{Slot}} \texttt{Zweifel}_{\texttt{Slot}} * (\texttt{Max}_{\texttt{importance}} - \texttt{importance}_{\texttt{Slot}})} > \zeta$$

Für jeden Slot des Nachrichtenelementes wird die Differenz der maximalen Wichtigkeit $\texttt{Max}_{\texttt{importance}}$ (der größte `importance`-Eintrag) und der `importance` des Elementes aufsummiert. Die Gesamtsumme wird durch die Summe der mit den Zweifelsfaktoren gewichteten Differenzen dividiert und sollte größer als ein heuristischer Wert ζ[13] sein. Dabei liegt der Zweifelswert erfolgreich instantiierter Slots ebenfalls bei "1". Wird ζ sehr niedrig angesetzt, genügen zur Akzeptanz eines *Nachrichtenelementes* die wichtigen Slots. Ist er zu hoch, so wird ein Element nur anerkannt, wenn alle wichtigen und sehr wichtigen Slots ohne verzögerte Regeln gebunden werden.

---

[13] Ein Wert der im Intervall]0,1[liegt und experimentell für das Nachrichtenmodell ermittelt werden sollte.

Instantiierung von Nachrichtentypen

Ein Nachrichtentyp gilt als nicht instantiiert, wenn das Vertrauensmaß $V_{Nachrichtentyp}$ zu niedrig ist. Es wird analog zum Vertrauensmaß der Nachrichtenelemente durch

$$V_{Nachrichtentyp} = \frac{\sum_{Nach.el.}(Max_{importance}\text{-}importance_{Nach.el.})}{\sum_{Nach.el.} Zweifel_{Nach.el.} * (Max_{importance}\text{-}importance_{Nach.el.})} > \zeta$$

definiert.

Instantiierung von Multi-Nachrichtentypen

Die Bewertung einer Instantiierung eines Multi-Nachrichtentypen hängt nur von dem Auftreteten eines sehr wichtigen Typs ab, z.B. dem, der die Aktivierung ins Leben gerufen hat. Alle anderen spielen keine Rolle, da sie optional sind.

3.5.2 Diskriminierungsfehler

Durch eine fehlerhafte Instantiierung, deren Ursachen im vorangegangenen Kapitel aufgelistet wurden, kann die Entscheidung für ein Nachrichtenelement während der Diskriminierung als falsch erkannt werden. Ohne daß dem Predictor durch die Nachrichtentypen weitere Informationen gegeben werden, muß ein Backtracking durchgeführt werden. Innerhalb der Struktur des beschnittenen Nachrichtenbaumes oder des Diskriminierungsbaumes kann es keine falsche Diskriminierung geben. Entweder ist das

Darstellung durch einen der Diskriminierungsbäume.

4 Schlußbemerkungen

Die Aufgabe des Predictors im Umfeld des Projektes ALV besteht darin, koordinierend die erwartungsgesteuerte partielle Analyse deutscher Briefdokumente zu leiten. Um eine Grundlage zu schaffen, auf dem ein erwartungsgesteuertes Analyseverfahren eingesetzt werden kann, wurde für die Domäne der Geschäftsbriefdokumente das Nachrichtenmodell entwickelt (siehe [Gores & Bleisinger 92]).

Der Predictor, die koordinierende und steuernde Komponente der Textanalyse, benutzt die zahlreichen Informationen, die durch das Nachrichtenmodell und die Umgebung in ALV bereitstehen. Dazu zählen vor allem das Lexikon, das durch die Bereitstellung vielfältiger lexical views die Wissensbasis neben dem Nachrichtenmodell darstellt. Der Einsatz eines erwartungsgesteuerten Analyseverfahrens für die Domäne der Geschäftsbriefe stellt ein geeignetes Verfahren zur partiellen Textanalyse dar. Insbesondere die Einbindung in ein Dokumentanalysesystem wie ALV bringt dabei große Vorteile, da durch die Layout- und Logikanalyse bereits ein umfangreiches Wissen zur Verfügung steht.

Um mit einem System dieser Art erfolgreich zu sein, muß zunächst das Nachrichtenmodell sehr sorgfältig modelliert werden. Dies erlaubt die gültigen, d.h. dem Modell entsprechende Dokumente, effizient zu analysieren und die gewünschte Information zu extrahieren. Zum derzeitigen Entwicklungsstand des Systems stehen dem Predictor nur wenige Substantiierer zur Verfügung, u.a. ein Klassifikator und ein Mustererkenner bzw. Schlüsselphrasensucher. Substantiierer, die auch syntaktische Strukturen erkennen können, sind zur Zeit noch nicht integriert. Zudem ist es wünschenswert, daß das Ausgabeverhalten der Substantiierer - soweit dies möglich ist - sich an dem Standard des Nachrichtenmodells orientiert, diese also CD-Formen, Nachrichtenelemente oder Nachrichtentypen als Ausgabe liefern können.

Die momentan eingesetzten Methoden, die dem Predictor zur Behebung von Analysefehlern zur Verfügung stehen, sind z.T. durch die Struktur des Nachrichtenmodells vorgegeben und können durch eine Erweiterung dessen erweitert werden. Dazu können z.B.

Für die zukünftige Weiterentwicklung der erwartungsgesteuerten partiellen Textanalyse ergeben sich folgende Anforderungen:

- Erweiterung des Modells hinsichtlich der Vollständigkeit, der Konsistenz und des Informationsgehaltes (insbesondere Steuerungsinformationen).
- Erweiterung des Predictors um eine subt[illegible]e Behandlung der Regeln und mehr Möglichkeiten, der Reaktion auf fehlgeschla[illegible]ne Substantiiereranfragen.

Die Fähigkeiten des bisher entwickelten Predictors zur Steuerung der Extraktion der wichtigen Informationen deutscher Geschäftsbriefdokumente im Zusammenspiel mit dem Nachrichtenmodell und den Substantiierern stellen einen ersten Schritt zur erwartungsgesteuerten partiellen Textanalyse dar. Durch unterschiedliche Erweiterungen, insbesondere den eben genannten, läßt sich jedoch eine Qualitätssteigerung der Ergebnisse der Analyse erreichen.

5 Index

6 Literatur

[Ali 92]
Majdi B. H. Ali: *Bildverarbeitungsroutinen für die Dokumentanalyse*, Projektarbeit, Universität Kaiserslautern, 1992.

[Becker 1975]
Joseph D. Becker: *The Phrasal Lexikon*; in R. Schank & B. L. Nash-Webber (eds.): *Theoretical Issues in Natural Language Processing* – An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics and Artificial Intelligence; June 10-13 1975, Cambridge Massachusetts.

[Bobrow et al. 77]
Daniel G. Bobrow, Roland M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson und Terry Winograd: *GUS, A Frame Driven Dialog System*; Artificial Intelligence 8, No. 1, 77, S. 155-173.

[Boon 92]
Josua Boon: *Ein wörterbuchbasiertes Texterkennungssystem als Teil eines Dokumenterkennungssystems*, Diplomarbeit, Universität Kaiserslautern, Juli 1992.

[Chomsky 56]
Noam Chomsky: *Aspects of the Theory of Syntax*, MIT Press, Cambridge MA, 1956.

[DeJong 79]
Gerald Francis DeJong II: *Skimming Stories in Real Time: An Experiment in Integrated Understanding*; Dissertation (Ph.D.), Faculty of the Graduate School of Yale University;Yale University, 1979.

[DeJong 82]
Gerald Francis DeJong II: *An Overview of the FRUMP System*; in Wendy G. Lehnert und Martin H. Ringle (eds.): Strategies for Natural Language Processing; Lawrence Erlbaum Associates, Hillsdale 1982.

[Dengel & Schweizer], Andreas Dengel und E. Schweizer, *Rotationswinkelbestimmung in abgetasteten Dokumentbildern*, in: H. Burckard, H. Höhne, B. Neumann (Hrsg.) Mustererkennung 1989 — Proceedings 11. DAGM-Symposium, Hamburg (Oktober 1989), Springer Verlag, Informatik Fachbericht 219, S. 274-278.

[Dengel 92]
A. Dengel. *ANASTASIL: A System for Low-Level and High-Level Geometric Analysis of Printed Documents*, in: H. Baird, H. Bunke, K. Yamamoto (eds.), Structured Document Image Analysis, Springer Publ. (1992).

[Dengel et al 92a]
A. Dengel, R. Bleisinger, R. Hoch, F. Hönes, F. Fein: *From Paper to Office Document Standard Representation*, IEEE Computer, July 1992.

[Dengel et al 92b]
A. Dengel, R. Bleisinger, R. Hoch, F. Hönes, F. Fein, M. Malburg: Π_{ODA}*: The Paper Interface to ODA*, DFKI Research Report RR-92-02, February 1992.

[Dengel et al 92c]
A. Dengel, A. Pleyer, R. Hoch. *Fragmentary String Matching by Selective Access to Hybrid Tries*. Proc. of 11th International Conference on Pattern Recognition, The Hague, August/September 1992.

[Dengel et al. 91]
Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Fein und Frank Hönes: *From Paper to an Office Document Standard Representation;* IEEE Computer, Juli 1992.

[Dittrich 92]
Stefan Dittrich: *Automatische, Deskriptor-basierte Unterstützung der Dokumentanalyse zur Fokussierung und Klassifizierung von Geschäftsbriefen;* Diplomarbeit, Universität Kaiserslautern, Mai 1992.

[Fein et al 92]
F. Fein, F. Hönes, B. Hornbruch: *Segmenting Business Letters - Problems and Solutions*, eingereicht bei 8. Scandinavian Conference on Image Analysis, Tromso, Norwegen, 1993.

[Fillmore, 1971]
C. Fillmore, *Types of Lexical Information*, in D.D. Steinberg & L.A. Jakobovits (eds.), *Semantics:An Interdisciplinary Reader*, Cambridge University Press, London, S. 370-392, 1971.

[Gores & Bleisinger 92]
Klaus-Peter Gores und Rainer Bleisinger: *Ein Modell zur Repräsentation von Nachrichtentypen*, DFKI-Document D-92-28, DFKI Kaiserslautern, Dezember 1992.

[Gores 1990]
Klaus-Peter Gores, *Ein bewertender Vergleich von Grammatikformalismen, Anwendungsmöglichkeiten in einem Dokumentanalysesystem*, Projektarbeit, Universität Kaiserslautern, 1990.

[Hayes 85b]
Philip J. Hayes, *Entity-Oriented Parsing*, Coling 1984.

[Hayes 88]
Philip J. Hayes, Laura E. Knecht and Monica J. Cellio:*A News Story Categorization System*, Proceedings of 2nd Conference on Applied Natural Language Processing, S. 9-17, Austin, Texas, Februar 1988.

[Hayes et al. 85a]
Philip J. Hayes, P. Andersen, S. Safier: *Semantic Caseframe Parsing and Syntactic Generality*, Proceedings of 23rd Annual Meeting of the Association for Computer Linguistics, S. 153-160, Chicago 1985.

[Hoch & Dengel 93]
Rainer Hoch, Andreas Dengel: *INFOCLAS: Classifying the Message in Printed, Business Letters*, Proceedings Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, USA, April 1993.

[Hoch & Malburg 92]
R. Hoch, M. Malburg. *Designing a Structured Lexicon for Document Image Analysis*. Proc. of Seventh Intl. Summer School on Information Technologies and Programming, 28 June - July 5, Sofia, 1992.

[Kamp 88]
H. Kamp: *Discurse Representation Theory: What it is and where it ought to go*, in A. Blaser: *Natural Language at the Computer*, Lecture Notes in Computer Science 320, Heidelberg 1988, zitiert in Christoph Klauck:*Diskursrepräsentation von Frage-Antwort-Dialogen auf Basis unifikationsbasierter Grammatikformalismen*, Diplomarbeit, Universität Kaiserslautern, 1990.

[Lebowitz 83]
Michael Lebowitz: *Memory-Based Parsing*, Artificial Intelligence **21** (4), S. 363-404, Oktober 1983.

[Lebowitz 85]
Michael Lebowitz: *An Experiment in Intelligent Information Systems: RESEARCHER*, 1985.

[Molter 92]
Michael Molter: *ZEBRA - Ein System zur Zeichenklassifikation für eingeschränkte Fontfamilien*, Diplomarbeit, Universität Kaiserslautern, August 1992.

[Rustin 73]
Randall Rustin (ed.): *Natural Language Processing*; Courant Computer Science Symposium No. 8; Algorithmics Press; New York 1973.

[Schank 72]
Roger C. Schank: *Conceptual Dependency: A theory of natural language understanding*, Cognitive Psychology, **3** (4), S. 552-631, 1972, zitiert in Encyclopedia of Artificial Intelligence.

[Schank 73]
Roger Schank: *The Conceptual Analysis of Natural Language;* in [Rustin 73], S. 291-309, zitiert in Encyclopedia of Artificial Intelligence.

[Schmidt 93]
Michael Schmidt: *Schlüsselwort-Substantiierer für die Dokumentanalyse,* Projektarbeit, Universität Kaiserslautern, Mai 1993.

[Stock &al 1989]
Oliviero Stock & Rino Falcone & Patrizia Insinnamo: *Bidirectional Charts, a Potential Technique for Parsing Spoken Natural Language Sentences;* Computer Speech and Language 3, 1989, S. 219-237; Academic Press.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen

DFKI Publications

The following DFKI publications or the list of all published papers so far can be ordered from the above address

RR-92-41
Andreas Lux: A Multi-Agent Approach towards Group Scheduling
32 pages

RR-92-42
John Nerbonne:
A Feature-Based Syntax/Semantics Interface
19 pages

RR-92-43
Christoph Klauck, Jakob Mauss: A Heuristic driven Parser for Attributed Node Labeled Graph Grammars and its Application to Feature Recognition in CIM
17 pages

RR-92-44
Thomas Rist, Elisabeth André: Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP
15 pages

RR-92-45
Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46
Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster:
WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47
Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48
Bernhard Nebel, Jana Koehler:
Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

RR-92-49
Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:
Heuristic Classification for Automated CAPP
15 pages

RR-92-50
Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51
Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52
Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-53
Werner Stephan, Susanne Biundo:
A New Logical Framework for Deductive Planning
15 pages

RR-92-54
Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55
John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56
Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58
Franz Baader, Bernhard Hollunder:
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59
Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
13 pages

RR-92-60
Karl Schlechta: Defaults, Preorder Semantics and Circumscription
19 pages

RR-93-02
Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist:
Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-03
Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi:
An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

RR-93-04
Christoph Klauck, Johannes Schwagereit:
GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

RR-93-05
Franz Baader, Klaus Schulz: Combination Techniques and Decision Problems for Disunification
29 pages

RR-93-06
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: On Skolemization in Constrained Logics
40 pages

RR-93-07
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: Concept Logics with Function Symbols
36 pages

RR-93-08
Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer: COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

RR-93-09
Philipp Hanschke, Jörg Würtz:
Satisfiability of the Smallest Binary Program
8 Seiten

RR-93-10
Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11
Bernhard Nebel, Hans-Juergen Buerckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12
Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13
Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14
Joachim Niehren, Andreas Podelski,Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-19

DFKI Technical Memos

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann: Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Busemann: Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang: Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh:
A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Arne Kilger: Realization of Tree Adjoining

DFKI Documents

D-92-12
Otto Kühn, Franz Schmalhofer, Gabriele Schmidt: Integrated Knowledge Acquisition for Lathe Production Planning: a Picture Gallery (Integrierte Wissensakquisition zur Fertigungsplanung für Drehteile: eine Bildergalerie)
27 pages

D-92-13
Holger Peine: An Investigation of the Applicability of Terminological Reasoning to Application-Independent Software-Analysis
55 pages

D-92-14
Johannes Schwagereit: Integration von Graph-Grammatiken und Taxonomien zur Repräsentation von Features in CIM
98 Seiten

D-92-15
DFKI Wissenschaftlich-Technischer Jahresbericht 1991
130 Seiten

D-92-16
Judith Engelkamp (Hrsg.): Verzeichnis von Softwarekomponenten für natürlichsprachliche Systeme
189 Seiten

D-92-17
Elisabeth André, Robin Cohen, Winfried Graf, Bob Kass, Cécile Paris, Wolfgang Wahlster (Eds.): UM92: Third International Workshop on User Modeling, Proceedings
254 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-92-18
Klaus Becker: Verfahren der automatisierten Diagnose technischer Systeme
109 Seiten

D-92-19
Stefan Dittrich, Rainer Hoch: Automatische, Deskriptor-basierte Unterstützung der Dokument-

D-92-23
Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24
Jürgen Müller, Donald Steiner (Hrsg.): Kooperierende Agenten
78 Seiten

D-92-25
Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26
Enno Tolzmann: Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27
Martin Harm, Knut Hinkelmann, Thomas Labisch: Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28
Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

D-93-01
Philipp Hanschke, Thom Frühwirth: Terminological Reasoning with Constraint Handling Rules
12 pages

D-93-02
Gabriele Schmidt, Frank Peters, Gernod Laufkötter: User Manual of COKAM+
23 pages

D-93-03
Stephan Busemann, Karin Harbusch(Eds.): DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

D-93-04
DFKI Wissenschaftlich-Technischer Jahresbericht

Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse

Klaus-Peter Gores, Rainer Bleisinger

D-93-07
Document