Domänenadaptive Kerntechnologien für die Analyse und Synthese natürlicher Sprache

Günter Neumann

LT-Lab, DFKI
Saarbrücken, Germany
neumann@dfki.de

Ausgewählte Schriften als Grundlage zur Erlangung der Habilitation im Fachgebiet Computerlinguistik an der Universität des Saarlandes

Inhaltsverzeichnis

Von B. Crysmann, A. Frank, B. Kiefer, H. Krieger, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, and F. Xu

Veröffentlicht in: Proceedings of ACL-2002, Philadelphia, July 2002, pp. 441-447.

Kapitel 1

Überblick

Die folgende Schriftensammlung umfasst acht Zeitschriften-, Buch- und Konferenzbeiträge, die ich in den letzten Jahren in den Forschungsgebieten *Natürlichsprachliche Systeme* ("Natural Language Systems", kurz NLS) und *Computerlinguistik* publiziert habe. Der rote Faden, der die Schriften verbindet ist in kompakter Form im Titel dieser Habilitation formuliert *Domänenadaptive Kerntechnologien für die Analyse und Synthese natürlicher Sprache*.

1.1 Einführung

Mit dem Aufkommen des Internets ist das Interesse an effizienter und robuster Sprachtechnologie enorm gestiegen. Sprachtechnologische Methoden spielen gegenwärtig eine große Rolle in verschiedenen Gebieten, wie zum Beispiel NL-basierte Dialog- und Auskunftssysteme, domänenoffene Frage-Antwort-Systeme, großangelegte ("large-scale"[1]) Informationsextraktion, Text Mining, Textzusammenfassung, Extraktion von domänenspezifischen Ontologien aus freien Texten oder wissensbasierte Suchmaschinen im Kontext des Semantic Webs. Dabei ist zu beobachten, dass eine NL–Verarbeitung nicht nur als wesentlicher Teil einer intelligenten Benutzerschnittstelle exploriert wird, sondern auch sehr stark in die Rolle getreten ist, Domänenwissen aus sehr großen Mengen von freien Texten zu extrahieren, d.h. wesentlicher Bestandteil eines datenorientierten Wissensaquisitionsprozesses geworden ist.

Unabhängig vom konkreten Einsatz sprachtechnologischer Methoden werden dabei stets dieselben Basisfunktionalitäten eingesetzt, soweit es die strukturelle Analyse betrifft, wenn auch die verschiedenen Aufgabenszenarien unterschiedliche Anforderungen verlangen. Salopp formuliert: Kasus bleibt Kasus, Subjekt Subjekt, egal ob im Kontext eines Dialogsystems oder eines intelligenten Informationsextraktionssystems. In allen diesen unterschiedlichen Anwendungen zeigt sich aber, dass die Art und Weise der Einbettung linguistischer Entitäten und Funktionalitäten in die domänenspezifische Objektwelt bei weitem nicht

[1] Diesen englischen Term werde ich im weiteren benutzen.

trivial ist, wenn nicht sogar den Flaschenhals für die Effektivität und den Nutzen von sprachtechnologischen Methoden darstellt.

Zur Erreichung einer effektiven Sprachverwendung müssen die linguistischen Resourcen optimal an die jeweilige Anwendung adaptiert werden. Es müssen dabei große Mengen von freien Texten schnell und robust verarbeitet werden können. Dabei spielt nicht nur die Größe der Textmengen eine Rolle (z.B. im Falle der intelligenten Informationsextraktion mehrere Megabyte und im Falle von domänenoffenen Frage–Antwort–Systemen prinzipiell das ganze Web), sondern die Tatsache, dass es sich um reale Texte handelt. Damit sind Phänomene, wie lexikalische und grammatikalische Lücken und Fehler, die Verwendung sehr langer Sätze, Telegram oder Email–Stil, Verwendung von Subsprachen sowie Kombination von freiem Text mit strukturierter Information — um nur einige zu nennen — mehr die Regel denn die Ausnahme.

In dieser Habilitationsschrift stelle ich einige in diesem Kontext von mir entwickelte Methoden und Algorithmen vor. Ich gehe hierbei von einer hybriden Systemarchitektur aus, die Aspekte aus den folgenden Bereichen modelliert:

- kompetenzbasierte Grammatikverarbeitung
- daten–und korpusgesteuerter Erwerb von Performanzgrammatiken
- Integration flacher und tiefer Sprachverarbeitung
- Intelligente Informationsextraktion (IE)

Bild 1.1 zeigt den Grundriss einer domänenadaptiven natürlichsprachlichen Kernarchitektur. Kernstück für die tiefe, kompetenzbasierte Grammatikverarbeitung ist ein uniformer Algorithmus zum Parsen (Analyse) und Generieren (Synthese) von natürlichsprachlichen Sätzen mit reversiblen constraint-basierten Grammatiken. Schwerpunkt hier ist die enge Verzahnung beider Verarbeitungsrichtungen als Grundlage zur Modellierung komplexer Verarbeitungsstrategien, wie systemimanente Selbstüberwachung und Revision, Paraphrasenauswahl oder kontrollierte Sprachproduktion.

Kernstück für eine flache Verarbeitung ist ein System zur effizienten und robusten Verarbeitung sehr großer, freier Textdokumente des Deutschen. Das System besteht aus einer Kaskade von sehr robusten modularen Komponenten. Auf allen Ebenen werden endliche stochastische Automatenmodelle uniform eingesetzt. Neben der geforderten sehr hohen Effizienz und Robustheit, liegt der Schwerpunkt auf Realisierung einer hohen, nachweisbaren linguistischen Abdeckung.

Die Integration der flachen und tiefen Sprachkomponenten, sowie die domänenspezifische Modellierung der IE–Anwendungen wird auf Ebene der strukturierten natürlichsprachlichen Objekte durchgeführt. Für die Spezifikation domänenadaptiver NL–Grammatiken kommen dabei auch maschinelle Lernverfahren zum Einsatz, die eine daten- und korpusgesteuerte Spezialisierung bzw. Induktion grammatikalischer Strukturen ermöglichen. Was die Domänenmodellierung eines IE–Kernsystems betrifft, so wird dies ebenfalls auf Basis eines automatischen Verfahrens zur Konstruktion eines IE–Modells (Lexikon, Extraktionsregeln und Ontologie) realisiert.

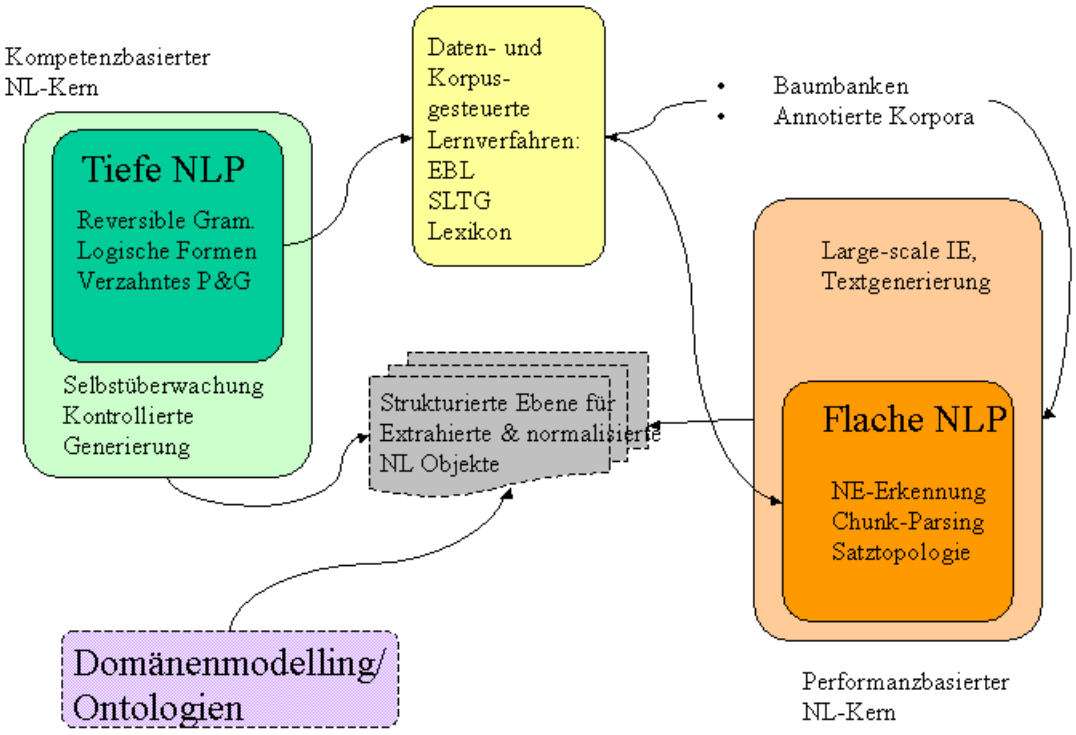

Abbildung 1.1: Der Entwurf eines domänenadaptiven natürlichsprachlichen Kernsystems

Es folgt nun eine kurze Beschreibung der zentralen Methoden und Algorithmen. Die Aufteilung in entsprechende Abschnitte folgt dem Grundriss der Architektur. In jedem Abschnitt werden dann Verweise auf die relevanten Schriften gemacht (gegebenenfalls mit Bemerkungen zur Autorenschaft), denen man die jeweiligen Details entnehmen kann.

1.2 Kompetenz–orientierte Kernkomponente für Parsing und Generierung

> "Interleaving Natural Language Parsing and Generation Through Uniform Processing", Günter Neumann, In Journal Artificial Intelligence 99, (1998) pp. 121-163.

In vielen Situationen des Sprachgebrauchs findet das Verstehen von Sprache und deren Produktion verzahnt statt. Zum Beispiel *überwachen* Menschen was sie sagen und wie sie es sagen. Sie planen und revidieren was sie sagen wollen, bevor sie es tatsächlich tun, z.B. um das Risiko zu vermeiden, missverstanden zu werden (was natürlich auch mit dem

Grad der Aufmerksamkeit des Sprechers korreliert). Oder sie versuchen die *kontrollierte* Generierung nicht–ambiger Äußerungen (vorausgesetzt die zugrunde liegende Botschaft ist es auch). Ein Sprecher ist in der Lage, die Verwendung seiner Sprache an den jeweiligen Hörer anzupassen (z.B. wenn neue, dem Hörer bisher unbekannte Sachverhalte mitgeteilt werden müssen, was allgemein in tutoriellen Situationen zwischen Experten und Novizen der Fall ist). Menschen sind auch recht geschickt in der *Vervollständigung* von Äußerungen, die vom Gesprächspartner begonnen wurden ("Ja, ich weiß was Sie sagen wollen: ..."), und sie sind recht effektiv im Auffüllen von Lücken in Äußerungen (z.B. zu beobachten bei psycholinguistischen Experimenten oder ähnlichen Sprachspielen).[2]

In dem obigen Artikel wird eine kompetenzbasierte Komponente für Parsing und Generierung vorgestellt, die die Modellierung solcher Aspekte der adaptiven Sprachverwendung ermöglichen bzw. vereinfachen soll. Die hierzu vorgelegte Schrift basiert auf meiner Dissertation von 1994, verbessert bzw. vereinfacht diese aber an zentralen Stellen. So definiere ich ein vereinfachtes, uniformes Indizierungsschema für die Verwaltung von Zwischenergebnissen, was eine Vereinfachung des gesamten Modells der Verzahnung von Parsing und Generierung ermöglicht. Die Kernkomponente basiert auf einem uniformen Deduktionsverfahren, dass gleichermaßen zur Analyse (Parsing) und Synthese (Generierung) mit reversiblen, constraint–basierten Grammatiken eingesetzt wird. Aufbauend auf diesem uniformen Deduktionsverfahren wird eine Methode zur engen Verzahnung von Parsing und Generierung vorgestellt und gezeigt, wie dies zur eleganten und einfachen Modellierung von domänen- und sprecheradaptiven Prozessen, wie z.B. Generierung eindeutiger Äußerungen oder Parsing und Generierung von Paraphrasen in interaktiven Frage–Antwort–Systemen eingesetzt werden kann.

Bemerkungen zur Aktualität Im Vergleich zu den anderen beigelegten Schriften gehört diese zu meinen älteren Arbeiten. Zu dem muss man eingestehen, dass mit dem verstärkten Aufkommen der sogenannten flachen Sprachtechnologien (vgl. 1.4.1) Anfang der 90iger Jahre, komplexe Performanzmethoden, wie z.B. Selbstüberwachung auf Basis eines verzahnten Modells bis Ende der 90iger Jahre eher von untergeordnetem Interesse waren, zumindest was die Realisierung von "large–scale" NL–Systemen betrifft. Allerdings lässt sich aktuell ein gesteigertes Interesse für die Thematik vom verzahnten Parsing und Generieren und Selbstüberwachung in recht unterschiedlichen Bereichen der Computerlinguistik erkennen.

So beschreibt [Kuh00] z.B. eine auf meinem verbesserten verzahnten Modell basierende Verarbeitungsstrategie von Syntax, die im Sinne der Optimalitätstheorie formuliert ist. Dort erlaubt die Verzahnung von Parsing und Generierung den Einsatz von generierungsbasierten Optimierungen während des Parsings.

An den englischen Universitäten Sussex und Brighton wurde aktuell das Gemeinschaftsprojekt COGENT (Controlled Generation of Text) mit einer Laufzeit von 2003–2006

[2]So gesehen kann Weizenbaum's ELIZA Programm auch als ein einfaches verzahnten Sprachmodell betrachtet werden: Für die Generierung einer Antwort werden auch Fragmente aus der Benutzereingabe verwendet und in eine ausgewählte Antwortschablone eingefügt, die im wesentlichen durch Vergleich mit Schlüsselwörtern aus der Eingabe bestimmt wird.

gestartet (vgl. [WCP+03]), in der die Idee der kontrollierten Generierung zum Zwecke einer domänenadaptiven Produktion von Äußerungen exploriert wird, wobei auch explizit auf meine Arbeiten im Bereich der kontrollierten Generierung von nicht–ambigen Äußerungen Bezug genommen wird. Weitere Anwendungen auf Basis der von mir vorgeschlagenen Methode des Self–Monitorings und des verzahnten Modells für die kontrollierte Generierung im Kontext von Dialogsystemen finden sich in [GKS02; KBdJW02; PO03]. In [HCW01] wird ein Ansatz beschrieben, in der Kontrollinformationen für die Generierung aus einer sehr großen (einfachen) Analysegrammatik unter Verwendung eines reversiblen Modells und Wiederverwendung eines statistischen Modells abgeleitet werden. Einen ähnlichen Ansatz habe ich auch Rahmen des BMBF–Verbundprojektes Getess realisiert und in [KBB+01] beschrieben.[3]

1.3 Regelbasierte Sprachmodelle mit HPSG

Was die Steigerung der Performanz des uniformen Modells betrifft, so war ich in erster Linie an Methoden interessiert, die eine domänenadaptive Spezialisierung grammatikalischer Strukturen erlauben (quasi eine domänen–gesteuerte Kompilation der Grammatik) ohne aber den Rückgriff bzw. die Integration mit der Kompetenzgrammatik zu blockieren. In den folgenden Abschnitten werden die hierzu relevanten Schriften kurz erläutert.

1.3.1 Motivation

Eine natürlichsprachliche Grammatik im Sinne der Kompetenz definiert grammatikalische Strukturen und Regelmäßigkeiten unabhängig von ihrer Verwendung in einer speziellen Domäne. Damit definiert sie aber auch eine sehr große Anzahl von theoretisch gültigen Analysen, die prinzipiell den gesamten Bereich an plausiblen linguistischen Konstruktionen abdeckt, inklusive sehr selten benutzter Entitäten. Daher erscheint eine Fokussierung auf Frequenz und Plausibilität von linguistischen Strukturen bezüglich einer speziellen Domäne als zumindest aus pragmatischen Gründen fruchtbarer Kompromiss für die Realisierung von "large–scale" NL–Systemen. Es wurden daher in den letzten Jahren eine Reihe von Methoden entwickelt, die es erlauben *automatisch* eine Kompetenzgrammatik für eine Domäne zu adaptieren. Eine auf diese Art spezialisierte Grammatik definiert einen bezüglich der Domäne optimalen grammatikalischen Suchraum und damit eine erhebliche Reduzierung der grammatikalischen Freiheitsgrade bzw. des Ambiguitätsgrades der Kompetenzgrammatik.

Sprachmodelle Allgemeiner betrifft es das Problem, ein für eine Domäne akkurates Sprachmodell automatisch abzuleiten. Ein Sprachmodell beschreibt i.a., welche Sequenzen von Wörtern in einer Domäne mit welchen Häufigkeiten vorkommen. Man kann zwei Arten von Sprachmodellen unterscheiden: statistische oder regelbasierte Modelle. Zumindest was

[3] Dort beschreibe ich eine Methode zum automatischen Erlernen parametrisierbarer Textschablonen für eine "flache" Generierung, die auf dem in Abschnitt 1.3.3 erläuterten regelbasierten Sprachmodell basiert.

eine einfache und schnelle Portierbarkeit auf neue Domänen betrifft, haben statistische Sprachmodelle den Nachteil, dass sie sehr große Mengen von Trainingsmaterial benötigen, die aber oft nicht vorhanden sind (bzw. deren Erstellung zu kosten- und zeitaufwendig ist). Gute regelbasierte Modelle können dagegen mit weit geringerem Trainingsmaterial erstellt werden.

Die zugrundeliegende Strategie zur Erstellung regelbasierter Sprachmodelle ist eine Variante des im Bereich des Maschinellen Lernens entwickelte Methode des *erklärungsbasierten Lernens* ("explanation-based learning", kurz EBL). Die zentrale Idee von EBL ist es, die von einem generischen Problemlösungsverfahren (z.B. einem Parser) für ein bestimmtes Problem (z.B. eine Wortsequenz) berechneten Ableitungsschritte (die in diesem Zusammenhang als *Erklärungen* interpretiert werden) zu generalisieren und in eine kompakte Form zu transformieren, die sehr effizient zur Lösung zukünftiger *ähnlicher* Probleme benutzt werden kann.

In der Computerlinguistik wurde EBL primär im Bereich des Parsings eingesetzt zur automatischen Adaption einer Kompetenzgrammatik für eine spezifische Domäne. Ausgehend von einer Kompetenzgrammatik, wird diese um ein domänenspezifisches Lexikon (welches auch Mehrwortlexeme und reguläre Grammatiken für Eigennamen umfassen kann) erweitert und in ein regelbasiertes Sprachmodell überführt. Die Kompilation wird durch einen Korpus gesteuert, der absolute und relative Häufigkeiten beisteuert. Das Sprachmodell kann dann in einem weniger komplexen Grammatikformat repräsentiert werden (z.B. als kontextfreie Grammatik), für das effiziente Parsingalgorithmen existieren. Ich konnte dann später zeigen, wie EBL auch im Bereich der Generierung von natürlicher Sprache zur Kompilation von domänenadaptiven Grammatiken (die im Falle der Generierung *kontrollierten* Sprachen entsprechen) effektiv eingesetzt werden kann, vgl. 1.3.3 und [KBB+01].

Einordnung meiner Arbeiten Betrachtet man die grammatikgesteuerte Analyse und Synthese von NL–Äußerungen als komplexen Suchprozess, wobei die Grammatik den potentiell unendlichen Suchraum implizit definiert, dann repräsentiert eine mittels der EBL–Methode erlernte Spezialgrammatik einen ausgezeichneten, domänenspezifischen Teilbereich des grammatikalischen Suchraums. Ausgangspunkt für EBL sind die Ableitungsbäume, die durch Analyse von Trainingssätzen mit der Kompetenzgrammatik bestimmt wurden. Ohne anschließenden Generalisierungsschritt beschreibt diese Menge der Ableitungsbäume einen endlichen, expliziten Suchraum.

Im gewissen Sinne handelt es sich um einen einfachen (dennoch strukturierten) Speicher von als wohlgeformt analysierte NL–Äußerungen. Zentrale Aspekte jeder EBL–Methode sind demnach a) die Wahl effizienter Indizierungsmechanismen und b) die Wahl der Generalisierungsmethode. Auf der einen Seite möchte man einen möglichst direkten Zugang von der Eingabe (z.B. einer Wortsequenz) auf die relevanten Teile des ausgezeichneten Suchraums und zum anderen möchte man aber auch durch Generalisierung eine möglichst breite Abdeckung von prototypischen Ableitungen erreichen. Unter diesen Gesichtspunkten habe ich die Ideen zur Erzeugung regelbasierter Sprachmodelle für spezielle Domänen aufgegriffen und erstmals im Zusammenhang mit HPSG–basierten Kompetenzgrammatiken

realisiert. Dabei waren wichtige Designkriterien:

- **Uniformität:** Unter der Annahme einer reversiblen Kompetenzgrammatik und eines uniformen Verfahrens für Parsing und Generierung wollte ich eine *uniforme* EBL–Methode realisieren, die zumindest potentiell ebenfalls für Parsing und Generierung eingesetzt werden kann.

- **Kompatibilität:** Es muss sichergestellt werden, dass die durch die EBL–Methode bestimmten Strukturen kompatibel sind mit denen der Kompetenzgrammatik. Damit wird es möglich, auf Ebene der strukturellen Beschreibung die Teilstrukturen aus den verschiedenen Verarbeitungsebenen ("flache" EBL–Methode versus "tiefe" uniforme Verarbeitung) zu integrieren.

- **Erweiterbarkeit:** Die Kompetenzgrammatik definiert für jede Spezialgrammatik, was strukturell möglich und erlaubt ist. Domänenspezifische Aspekte werden im wesentlichen durch lexikalische Entitäten eingeführt. Allerdings setzt dies voraus, dass die Kompetenzgrammatik umfassend alle möglichen strukturellen Variationen einer Sprache kodiert. Solange dies jedoch nicht der Fall ist (und die Entwicklung von "very large–scale" Kompetenzgrammatiken ist ja noch ein aktives Forschungsgebiet), müssen alternative Weg gefunden werden, wie grammatikalische Lücken in der Spezialgrammatik gefüllt werden können. Eine Möglichkeit besteht, manuell erstellte Baumbanken (wie z.B. die Penn Treebank oder das Negra Korpus) zu verwenden und diese ebenfalls per EBL zu kompilieren. Damit bestünde die Möglichkeit, kompetenzbasierte EBL–Grammatiken mit "baumbankenbasierten" EBL–Grammatiken zu kombinieren.

In [Neu94] habe ich bereits einen ersten Vorschlag für den Einsatz von EBL zur automatischen Extraktion von Subgrammatiken aus HPSG–Quellgrammatiken gemacht. In einem technischen Bericht (vgl. [Neu97]) habe ich darüber hinaus bereits eine inkrementelle EBL–Methode vorgeschlagen, in der Trainings- und Anwendungsphase eng verzahnt werden können. Ich konnte dann später aufgrund der Verfügbarkeit großer HPSG–Grammatiken (speziell derjenigen, die am CSLI der Stanford Universität im Projekt Lingo entwickelt wurden) meine ersten Entwürfe erheblich verbessern bzw. neuartige Methoden entwickeln, die ich auch ausführlich evaluiert habe. Die folgenden drei Schriften thematisieren und explorieren diese Arbeiten im Detail. Ich möchte daher die wesentlichen Aspekte hier nur kurz skizzieren.

1.3.2 Daten–orientiertes Parsing mit HPSG

> "Data-driven Approaches to Head-driven Phrase Structure Grammar", Günter Neumann, angenommen zur Veröffentlichung in Rens Bod, Remko Scha and Khahil Sima'an (eds): Introduction to Data-oriented Parsing. CSLI publications, Stanford, Ca, erscheint 2003, 21 Seiten.

In diesem Artikel wende ich die Idee der daten–gesteuerten Spezialisierung von HPSG–Grammatiken an, umso eine effiziente und anwendungsrelevante Verarbeitung von HPSG–basierten Grammatiken zu erreichen. Neben einer Steigerung der Effizienz sind auch folgende Aspekte meiner Methode (die ich kurz auch als HPSG–DOP bezeichne) relevant: 1) die Strategien zur Generalisierung der Spezialgrammatik basieren unmittelbar auf formalen Eigenschaften des HPSG–Formalismus, 2) die gesamte Lernstrategie soll einfach und transparent formuliert sein, sodass diese Methode auch ohne Kenntnisse der Details eingesetzt werden kann, und 3) die abgeleitenen Spezialgrammatiken sollen ein hohes Mass an Deklarativität besitzen, um so die oben genannten Designkriterien zu erfüllen.

HPSG–DOP extrahiert automatisch eine stochastische, lexikalisierte Baumgrammatik ("stochastic Lexicalized Tree Grammar", kurz SLTG) aus einer HPSG Quellgrammatik und einem repräsentativen Korpus von domänenrelevanten Äußerungen. Die Verarbeitung einer SLTG wird mit einem sehr schnellen spezialisierten Parser durchgeführt. Die Methode wurde ausführlich getestet mit der oben genannten "large-scale" Lingo–Grammatik und Korpora aus der VerbMobil–Domäne. Die hier vorgestellte Methode hat folgende Vorteile:

- Eine SLTG beschreibt eine kontextfreie Subsprache und erfüllt die Bedingungen einer lexikalisierten kontextfreien Grammatik.
- Es ist garantiert, dass keine Information (modulo Abdeckung) der Quellgrammatik verloren geht (inklusive der semantischen Beschreibungsebene der Grammatik).
- Da die Trainingsphase durch die HPSG–Prinzipien gesteuert wird, erreicht der gesamte Ansatz die erwünschte Einfachheit und Transparenz.
- Unser Ansatz erlaubt die systematische Integration von statistischer Information mit HPSG–Strukturen, was die Formulierung von präferenzbasierten, robusten Verarbeitungsstrategien ermöglicht.
- Der gesamte Ansatz hat ein sehr hohes Anwendungspotential, z.B. für die Verarbeitung kontrollierter Sprachen oder Informationsextraktion, wobei reichhaltige und linguistisch motivierte Informationen von HPSG zur Verfügung stehen.

Darüberhinaus kann dieselbe Methode auch für die Generierung eingesetzt werden (vgl. Abschnitt 1.3.3) und mit Grammatiken, die aus Baumbanken abgeleitet wurden (vg. 1.3.4) verknüpft werden. Daher kann man das zugrunde liegende Lernverfahren auch als performanzorientierte Erweiterung meines bereits dargelegten uniformen Modells auffassen (vgl. 1.2).

1.3.3 Daten–orientiertes Generieren mit HPSG

> "Applying Explanation-based Learning to Control and Speeding-up Natural Language Generation", Günter Neumann, in Proceedings of ACL-97, Madrid, 1997, pp. 214-220.

Eine Spezialgrammatik kann als Beschreibung einer domänenspezifischen Menge prototypischer Konstruktionen aufgefasst werden. Daher ist der EBL–Ansatz auch für die natürlichsprachliche Generierung (NLG) von großem Interesse. Allgemein wird der NLG–Prozess als ein komplexer — eventuell kaskadierter — Entscheidungsprozess modelliert. Dabei wird ein NLG–Gesamtsystem grob in zwei Kernkomponenten aufgeteilt: 1) die strategische Komponente, in der entschieden wird, *was* gesagt wird, und 2) die taktische Komponente, in der entschieden wird, *wie* das Ergebnis der strategischen Komponente in natürlicher Sprache formuliert wird. Die Eingabe für die taktische Komponente ist im wesentlichen eine semantische Repräsentation, die von der strategischen Komponente berechnet wurde. Unter Verwendung eines Lexikons und einer Grammatik, ist deren Hauptaufgabe, die Berechnung potentiell aller möglichen Sätze, die mit der semantische Eingabe kompatibel sind. (In einem konkreten Anwendungsszenarium muss das NLG dann die möglichst beste (z.B. minimal ambige) Äußerung auswählen.)

Nun kann die EBL–Methode in dem gleichen Sinne, wie sie für das Parsing zur Kontrolle der Menge der erlaubten Wortketten und des Ambiguitätsgrades eingesetzt wird, im Falle der Generierung benutzt werden, um die Menge der erlaubten semantischen Eingaben und den Grad an *Paraphrasierung* zu kontrollieren.[4] Der zentrale Vorteil für ein NLG–System ist der, das die Komplexität des (linguistisch–orientierten) Entscheidungsprozesses erheblich reduziert wird, da die EBL–Methode eine Adaption eines NLG–Systems an eine spezifische Sprachverwendung ermöglicht. Bezogen auf den Aspekt der Uniformität und Reversibilität ist folgendes wichtig:

- Zentraler Unterschied zur EBL–Parsing–Methode ist die Art der Indizierung der generalisierten Ableitungsbäume. Und dies betrifft auch nur die Vorverarbeitung bzw. Normalisierung der semantischen Eingabestrukturen (es handelt sich hierbei um Ausdrücke im Formalismus der Minimal Recursive Semantics, vgl. [CFS97]). Die normalisierten Ausdrücke können dann wie Wortketten betrachtet und daher mit den entsprechenden Algorithmen der EBL–Parsing–Methode verarbeitet werden.

- Ausgangspunkt für das Generierungsverfahren sind ebenfalls die Ableitungsbäume einer Trainingsmenge. Für das Lernverfahren spielt es dabei keine Rolle, ob diese Ableitungsbäume nun durch Parsing oder Generierung bestimmt wurden (diese Information kann natürlich für eine intelligente Speicherverwaltung berücksichtigt werden).

1.3.4 Ein uniformes Verfahren zur automatischen Extraktion von lexikalisierten Grammatiken aus Baumbanken und HPSG

"A Uniform Method for Automatically Extracting Stochastic Lexicalized Tree Grammars from Treebanks and HPSG", Günter Neumann, angenommen zur

[4]Das von mir vorgeschlagene maschinelle Lernverfahren ist übrigens auch eines der ersten korpusbasierten Grammatikverfahren, dass im Forschungsbereich Natürlichsprachliche Generierung überhaupt vorgeschlagen wurde, vgl. hierzu auch http://www.dcs.shef.ac.uk/~kalina/ml-stats-nlg-bibliography.html

Veröffentlichung in A. Abeille (ed): Building and using syntactically annotated corpora, Language and Speech series, Kluwer, erscheint, 2003, 16 Seiten.

Zentraler Aspekt in dieser Schrift ist die Übertragung des Verfahrens zur Extraktion einer stochastisch lexikalisierten Baumgrammatik (SLTG) für HPSG (vgl. 1.3.2) auf *Baumbanken*. Baumbanken sind manuell annotierte freie Textdokumente (z.B. Nachrichtentexte bekannter Agenturen oder Zeitschriften). Die Annotationen umfassen dabei Satzgrenzen, Wortarten, Syntax und oberflächennahe Semantik. Von besonderem Interesse für das SLTG–Verfahren ist die Tatsache, dass es sich um große Mengen von annotiertem Material handelt (z.B. umfasst die Penn Treebank ca. 1.000.000 annotierte Wörter, das Negra Korpus ca. 350.000) und das es sich hierbei um freie Texte handelt, mit teilweise sehr langen Sätzen (sehr oft 40 und mehr Wörter). Damit bieten die Baumbanken ein realistisches Szenarium, um die Effektivität von Lernverfahren zu untersuchen.

Die initiale Motivation zur Beschäftigung mit Baumbanken verdanke ich der Tatsache, dass am Anfang dieser meiner Forschungsarbeit nur mittelgroße Kompetenzgrammatiken zur Verfügung standen, mit denen man in der Regel auch nur kurze Sätze in vernünftiger Zeit analysieren konnte. Da meine Verfahren aber von Beginn an so angelegt waren, dass die Trainingsphase auf einer Menge von Ableitungsbäumen operiert, war es nahe liegend, diese Menge von Ableitungsbäumen quasi als dynamisch erzeugte Baumbank zu interpretieren. Diese Sicht der Dinge erlaubte es mir dann, das ursprünglich für HPSG entworfene SLTG–Verfahren auch auf große, manuell erstellte Baumbanken anzuwenden, wobei im konkreten Falle die Penn Treebank und das Negra Korpus zur Verfügung standen.[5] Wesentliche Unterschiede ergaben sich aber aus der Tatsache, dass den Baumbanken nur eine rudimentäre linguistisch explizite Formalisierung zu Grunde liegen. So existiert im Gegensatz zum HPSG–Verfahren keine explizite Formulierung eines Head oder Modifier Prinzipes. Daher musste ich diese entsprechenden linguistischen Prinzipien manuell durch Heuristiken simulieren, was im Falle des Negra Korpus allerdings weniger kompliziert war, da hier eine Annotation gemäß dependenztheoretischer Aspekte vorgenommen ist.

Im Unterschied zum HPSG–Verfahren erlaubt das Baumbank-Verfahren die Extraktion von lexikalisierten Bäumen mit multiplen lexikalischen Ankern. Damit sind die Baumbank SLTG–Grammatiken selektiver bezüglich lexikalischer Kollokationen als HPSG–basierte SLTG–Grammatiken, da dort nur ein lexikalischer Anker pro elementarer Baum erlaubt ist. Von zentralem Interesse ist auch die Tatsache, dass die extrahierten Baumbank–Grammatiken ebenfalls im gleichen Format deklarativ repräsentiert sind, wie ihre HPSG–basierten "Schwestern". Dies ist aber eine wichtige Grundlage für Verfahren zur Integration der verschiedenen SLTG–Grammatiken. In der oben aufgeführten Schrift erläutere ich die Grundzüge eines Verfahrens, wie eine Baumbank und eine HPSG–Grammatik auf Ebene des SLTG–Verfahrens integriert werden können.

[5]Die elementaren Bäume einer SLTG (unabhängig davon, ob als Basis eine HPSG oder eine Treebank verwendet wurde) haben formal und auch inhaltlich eine große Nähe zu den elementaren Bäumen lexikalisierter Baumadjunktionsgrammatiken ("lexicalized tree adjoining grammars, LTAGs"). Daher war es nicht überraschend (zumindest wenn man den Zeitpunkt betrachtet), dass EBL–basierte Verfahren und speziell die SLTG–basierten Lernverfahren auch im Forschungsbereich "Baumadjunktionsgrammatiken" sehr an Interesse gewonnen haben, vgl. [Xia99; CVS00; Chi00; XP00].

Solch eine Art der Integration bietet eine Menge von Vorteilen (neben den bereits oben aufgeführten Aspekten), insbesondere was die Erforschung neuartiger induktiver Lernverfahren betrifft. Eine mögliche Strategie wäre, mit einer relativ kleinen HPSG–basierten SLTG–Grammatik (eine sogenannte "seed grammar") zu beginnen und ausgehend von als domänenrelevant erkannten Textkorpora (hierzu können z.B. bekannte Textklassifikationsverfahren eingesetzt werden) eine immer größer werdende SLTG–Grammatik zu induzieren. Hierbei wäre es unter Einsatz von evolutionären Strategien möglich, automatisch mehr oder weniger "beliebige" Ableitungsbäume zu erzeugen, deren Wohlgeformtheit (oder "Fitness") durch Anwendung der HPSG–Prinzipien überprüft werden könnten. Natürlich handelt es sich hierbei nur um einen möglichen, zukünftigen Forschungsaspekt mit dem inhärenten Risiko des Fehlschlages. Im positiven Falle würde sich aber ein sehr hohes Anwendungspotential für die Zukunft ergeben, z.B. eine durch das Anwendungssystem stärker eigenständig kontrollierte Domänenadaption.

1.4 Kerntechnologien für Intelligente Informationsextraktion

1.4.1 Überblick

In den folgenden Schriften stelle ich innovative Methoden und Technologien vor, die ich im Bereich der intelligenten Informationsextraktion (IE) entwickelt habe. IE–Systeme zeichnen sich dadurch aus, dass sie gezielt aufgabenspezifische Information aus elektronischen Texten aufspüren und strukturieren können, bei gleichzeitigem "überlesen" irrelevanter Information. Das zentrale Ziel ist hierbei die Entwicklung von hochadaptiver und modularer IE–Kernfunktionalität auf der Basis von leistungsstarken, robusten und effizienten natürlichsprachlichen Komponenten und objektorientierten Werkzeugen zur Erstellung domänenspezifischer Wissensquellen, die in einfacher und dennoch flexibler Weise für IE–Anwendungen unterschiedlicher Domänen und Komplexität konfigurierbar sind.

Informationsextraktion Die Kernfunktionalität eines IE–Systems lässt sich kurz wie folgt charakterisieren: Auf der Basis einer vorgegebenen Spezifikation der relevanten Information in Form von getypten Attribut–Wert Matrixen, auch Templates genannt (z.B. Unternehmensbeteiligungen, Umsatzmeldungen, Gewinnentwicklungen, Produktinformation oder Personalwechsel im Managementbereich) und einer Menge von domänenspezifischen Texten isoliert das IE-System die relevanten Textfragmente, extrahiert aus jedem Fragment die spezifisch erwünschte Information und konstruiert aus den verschiedenen Informationsteilen in systematischer Weise die komplexe Zielstruktur. Dieser Prozess wird auch als Templateinstantiierung bezeichnet.

Folgendes Beispiel soll das gerade Erläuterte illustrieren. Wir betrachten die Aufgabe, Informationen über Personalwechsel aus Online-Dokumenten zu extrahieren. Insbesonde-

re wollen wir wissen, welche Person (*PersonOut*[6]) welche *Position* welcher *Organisation* wann (*TimeOut*) verlassen hat, und welche neue Person (*PersonIn*) wann (*TimeIn*) diese *Position* wieder aufgenommen hat. Das dazugehörige Template hat folgende Form:

$$[PersonOut\ PersonIn\ Position\ Organisation\ TimeOut\ TimeIn]$$

Für den folgenden Text:

> Dr. Hermann Wirth, bisheriger Leiter der Musikhochschule München, verabschiedete sich heute aus dem Amt. Der 65jährige tritt seinen wohlverdienten Ruhestand an. Als seine Nachfolgerin wurde Sabine Klinger benannt. Ebenfalls neu besetzt wurde die Stelle des Musikdirektors. Annelie Häfner folgt Christian Meindl nach.

ergibt sich dann beispielsweise das gefüllte (instanziierte) Template:

PersonOut	Dr. Hermann Wirth
PersonIn	Sabine Klinger
Position	Leiter
Organization	Musikhochschule München
TimeOut	heute
TimeIn	

Hier ist zu beachten, dass die gesamte Information über zwei Sätze verteilt ist, wobei im zweiten relevanten Satz sogar ein anaphorischer Ausdruck aufgelöst werden muss ("seine Nachfolgerin"). Wenn wir eine satzorientierte Verarbeitung annehmen, in der in einem ersten Schritt die relevante Information pro Satz bestimmt wird, dann müssen in einem nachfolgenden Verarbeitungsschritt die verschiedenen Teile zusammengefügt werden. Darüberhinaus ist der konkrete Wert für das Attribut *TimeOut* im Text nur relativ genannt ("heute"). Der Wert für das Attribut *TimeIn* ist explizit nicht genannt, so dass er nur über zusätzliche, eventuell sehr spezifische Annahmen ableitbar wäre, was wir für das Beispiel nicht tun wollen. Da nicht alle Attribute mit Werten belegt werden können, spricht man auch von einer *partiellen* Instanz.

Die so extrahierten und normalisierten Daten können dann vielseitig eingesetzt werden, z.B. zur feinkörnigen Textfilterung oder -klassifikation, als Einträge für Datenbanken oder Informationsmanagementsysteme, oder zur aufgabenspezifischen und inhaltsbasierten Navigation in on-line-Textdokumenten.

Anwendungsmodellierung In der Regel beginnt die Spezifikation einer Anwendung mit der Definition der domänenspezifischen Templatestruktur und der Definition eines Domänenlexikons, das Beziehungen zwischen Templates und relevanten Wörtern festlegt. In beiden Bereichen werden verstärkt objektorientierte Ansätze eingesetzt, um den Grad

[6]Die Attribute sind kursiv hervorgehoben.

an potentieller Wiederverwendung zu erhöhen. Zur Erkennung und Extraktion relevanter Phrasen (Wortsequenzen) werden domänenspezifische Grammatikregeln definiert. Angewandt auf aktuelle Dokumente liefern diese Phrasenregeln potentielle Slotfiller, die mittels domänenspezifischen Regeln zu instantiierten Templates kombiniert werden.

Die Entwicklung der unterschiedlichen Wissensquellen wird auf der Basis eines mit den Zielstrukturen (Templates) markierten Textkorpus durchgeführt. In der Regel werden von diesem Korpus 10% während des Entwicklungszyklus unberücksichtigt gelassen, um diesen Anteil später zur Evaluation der potentiellen Erkennungsrate zu verwenden.

Flache Sprachtechnologie Das hohe Mass an Robustheit und Effizienz, dass für eine Verarbeitung großer Textmengen erforderlich ist, wird durch Einsatz *flacher* Sprachtechnologie erreicht. Im Unterschied zu den tiefen, kompetenzbasierten Sprachtechnologien werden hierbei bestimmte generische Sprachregularitäten, von denen bekannt ist, dass sie Komplexitätsprobleme verursachen, entweder nicht oder ganz pragmatisch behandelt, z.B. durch Beschränkung der Rekursionstiefe auf Basis einer Korpusanalyse oder durch Verwendung von Heuristiken ("präferiere längstmögliche Teilketten"). Um mit dem durch die Internet–Revolution ausgelösten enorm steigenden Bedarf an Sprachtechnologie schritthalten zu können, arbeiten Forscher gerade im Bereich der IE an Methoden, die einen Kompromiss zwischen theoretischen Ansprüchen und pragmatischen Anforderungen darstellen.

Diese ingenieursmäßige Sichtweise auf Sprachverarbeitung hat zu einer Renaissance und Weiterentwicklung von bekannten Technologien geführt, insbesondere von endlichen Zustandsautomaten oder Transduktoren in der Syntaxanalyse. Im Unterschied zu endlichen Automaten (die ein Erkennen von regulären Mustern in einer Eingabesequenz erlauben) definieren Transduktoren zusätzlich eine Relation zwischen möglichen Sequenzen von Eingabesymbolen (z. B. Wörtern und ihren zugeordneten lexikalischen Merkmalen) und zugeordneten Ausgabestrukturen (z.B. Phrasen in Form einer Dependenzstruktur). Obwohl die Syntax einer natürlichen Sprache mindestens kontextfrei ist, zeigen aktuelle Arbeiten im Bereich der IE, dass bereits mit den einfacheren Transduktoren sehr praktikable Systeme realisierbar sind.

Die Tatsache, dass in aktuellen IE–Systemen in der Regel nur (kaskadierte) reguläre Grammatiken eingesetzt werden, hängt stark damit zusammen, dass meist nur einfache Instanzen von Relationen zu extrahieren sind. Dies und die Tatsache, dass große Mengen von freien Texten in der Regel nur partiell analysiert werden können (wegen lexikalischer und grammatikalischer Lücken und Fehler, sehr großen Satzlängen, nicht–textueller Information gemischt mit textueller Information usw.) hat zur Entwicklung von kaskadierten Fragment–Parsern geführt, in denen z.B. zuerst alle potentiell interessanten Nominalphrasen und Verbgruppen identifiziert werden, bevor versucht wird, diese mittels domänenspezifischen Satzmustern zu Instanzen von Relationen zu verknüpfen.

Allerdings hat sich gezeigt, dass für mehrstellige Relationen (sogenannte "scenario templates") diese Art der Fragmentanalyse eine obere Operationalitätsschranke von 60% F–

Maß zu haben scheint, vgl. [AI97].[7] Wir konnten in unseren Arbeiten darüber hinaus zeigen, dass für eine fragmentarische Syntaxanalyse von Sprachen mit freier Wortstellung, eine reine bottom–up Strategie sogar im Falle von einfachen Templatestrukturen nicht die nötige Robustheit besitzt, sondern zumindest die Erkennung der topologischen Satzstruktur erfordert, vgl. [NBP00]. Dies hat uns dann zur Erkenntnis verholfen, dass eine Integration von flacher und tiefer Verarbeitung (u.a. auch mittels der oben erläuterten Methoden, vgl. 1.3) helfen könnte, eine der Komplexität der Domänen und Aufgaben angepasste, parametrisierbare Verstehenstiefe (quasi eine Art "vom Interesse geleitetes Zoom–Parsing") zu modellieren. Wir werden in der letzten Schrift eine Systemarchitektur vorstellen deren Novum es ist, eine Plattform für die Integration von flachen und tiefen Verarbeitungskomponenten bereitzustellen.

Einordnung meiner Arbeiten Die in den Schriften vorgestellten IE–Kernsysteme wurden im wesentlichen im Rahmen von zwei BMBF–geförderten DFKI–Projekten durchgeführt: Paradime (Parametrizable Domain-adaptive Information and Message Extraction, Laufzeit: 1997–2000) und Getess (German Text Exploitation and Search System, Laufzeit: 1998-2001).[8] Als Projektleiter war ich für die inhaltliche und technische Realisierung zuständig. Die Konzeption, inhaltliche Ausarbeitung und ein Großteil der Implementation der Kernalgorithmen wurde von mir durchgeführt. Wie es aber im Rahmen der Entwicklung von Systemen in anwendungorientierten Forschungsprojekten die Regel ist, wurde die vollständige Umsetzung in Gruppenarbeit geleistet. Die Reihenfolge der Autoren in den ausgewählten Schriften spiegelt dabei primär die Arbeitsanteile an der jeweiligen Veröffentlichung wieder, d.h. die ersten beiden Schriften wurden primär von mir initiiert und durchgeführt[9], wogegen die dritte Schrift (vgl. 1.4.3) mit gleichem Aufwand von den Autoren erstellt wurde.

1.4.2 Linguistische Kerntechnologie

In den folgenden Schriften

> "An Information Extraction Core System for Real World German Text Processing", G. Neumann, R. Backofen, J. Baur, M. Becker, C. Braun, in: Proceedings of 5th ANLP, Washington, 1997, pp. 209-216.

und

> "A Shallow Text Processing Core Engine", G. Neumann und J. Piskorski, in: Journal of Computational Intelligence, Volume 18, Number 3, 2002, pages 451-476.

[7] Das F–Maß ist eine Beziehung zwischen Präzision und Vollständigkeit. Der angegebene Wert von 60% ist ein Erfahrungswert.

[8] Da es sich bei Getess um ein Verbundprojekt handelte, beziehe ich mich hierbei auf das DFKI–Teilprojekt.

[9] Was die erste Schrift betrifft, so habe ich diesen Text tatsächlich alleine verfasst, die anderen Autoren aber mit aufgeführt, da sie wesentlich bei der Implementation des Systems SMES mitgewirkt haben.

wird die linguistische Kerntechnologie zur Verarbeitung speziell deutschsprachiger elektronischer Texte beschrieben (die im weiteren mit *LT–Kern* abkürzend benannt wird). In der darauf folgenden Schrift wird eine Methode zur inkrementellen Erstellung eines domänenspezifischen, ontologiebasierten IE–Systems vorgestellt, dass auf dem LT–Kern aufbaut und diesen mit einer Ontologie–Werkbank kombiniert.

Das zentrale Ziel bei der Entwickling des LT–Kerns ist hierbei die einfache Portabilität und Adaptierung für IE-Anwendungen unterschiedlicher Domänen und Komplexität (d.h. unterschiedlich "tiefer" Verstehensleistung). Dies wird durch Bereitstellung von leistungsstarken, robusten und effizienten natürlichsprachlichen Komponenten und linguistischen Wissensquellen erreicht, die in einfacher und dennoch flexibler Weise für unterschiedliche Aufgaben konfigurierbar sind. Während der Entwicklung der Basistechnologie wurden hierbei sehr innovative sprachtechnologische Methoden entwickelt, u.a.

- hochabdeckende Eigennamenerkennung auf Basis robuster Mustererkennung (z.B. Namen von Personen, Firmen, Organisationen, Orte, sowie spezielle Datums- und Zeitausdrücke)
- sehr effiziente und robuste Analyse von deutschen Komposita
- hochabdeckende Analyse der topologische Struktur deutscher Sätze (vgl. auch [NBP00])
- robuste Erkennung unterspezifizierter Dependenzstrukturen und grammatikalischer Funktionen

und auch ausführlich evaluiert (neben den dieser Habilitation beigelegten Schriften vergleiche man auch [DKN98; KDN98; NBP00; NS02]). Darüber hinaus stellen wir in [DN01] ein auf Basis des LT–Kerns realisiertes Verfahren zur kaskadierten, flachen Referenzresolution vor.

Die hier beschriebene LT–Kerntechnologie hat ein hohes Applikationspotential und wurde bereits in vielen verschiedenen Anwendungsbereichen eingesetzt, u.a. Verarbeitung von Email–Nachrichten ([BDD+97]), Textklassifikation ([NS02]) Verteilung von Texten in einem Call Center, Text Mining, Extraktion von Firmeninformationen (diese im Rahmen von industriellen Anwendungen), Text Clustering ([HMS01]), hybride Lernverfahren von Relationen ([Mor02]) und intelligente Informationsextraktion auf der Basis integrierter Ontologien ([KBB+01]). Darüber hinaus wurde die Technologie in einer Reihe von externen Diplomarbeiten erfolgreich eingesetzt, u.a. [Vol00; Asc01; Eul01; Hei01].

In den meisten genannten Arbeiten wurde die Domänenmodellierung der IE–Anwendungen in der Regel durch flache Templatestrukturen realisiert. Im Kontext des BMBF–Verbundprojektes Getess hat sich aber gezeigt, dass bei zunehmender Komplexität der Templatestrukturen bis hin zur Beschreibung von Domänen mittels Ontologien, komplexere Prozesse der Domänenmodellierung nötig werden. Im nächsten Abschnitt will ich einen möglichen Ansatz kurz beschreiben, den ich zusammen mit Kollegen des AIFB (einem der Projektpartner) im Rahmen des Getess–Projektes realisiert habe.

1.4.3 Integration mit Domänenwissen

> "Bootstrapping an Ontology-based Information Extraction System", A. Mädche, G. Neumann und S. Staab, in: Szczepaniak, Segovia, Kacprzyk and Zadeh (eds), Intelligent Exploration of the Web, Springer, ISBN 3-7908-1529-2, 2002, pp. 345-360.

Der zentrale Flaschenhals für die gegenwärtige IE–Technologie liegt in der adäquaten Spezifikation des lexikalischen Wissens, der domänenspezifischen Extraktionsregeln und der Ontologie (zusammen bezeichnen wir dies als *IE–Modell*), sodass ein IE–System gleichermassen eine hohe Abdeckung und Präzision erreichen kann.

Zur Zeit wird bei der Entwicklung von IE–Systemen ein daten–gesteuerter Ansatz präferiert, der auf einer sehr genauen Untersuchung relevanter Textkorpora basiert.[10] In früheren Systemen (wie z.B. dem Fastus–System, vgl. [AHB+93]) wurde die Korpusanalyse und die Spezifikation der Extraktionsregeln manuell durchgeführt. Es wurde aber sehr bald deutlich, dass dies eine sehr zeitaufwendige Arbeit ist. Es zeigte sich aber auch, dass die Verwendung domänenunabhängiger NL–Komponenten (dies entspricht unserem LT–Kern) hilft, den Aufwand der Adaption eines IE–Systems auf die domänenspezifischen Aspekte erheblich zu minimieren.

Gleichzeitig hat die Entwicklung von modularen und wiederverwendbaren (da domänenunabhängigen) NL–Komponenten auch die Entwicklung von IE–spezifischen maschinellen Lernverfahren vorangetrieben. Diese Lernverfahren induzieren Teile des Domänenmodells (z.B., Regeln zum Füllen einzelner Slots oder einfache binäre Relationen über Eigennamen) aus verschiedenen annotierten oder nicht–annotierten Textkorpora, vgl. [CM98; YGTH00; RY02; PP03].[11]

Was allerdings bisher fehlt ist ein integrierter maschineller Lernansatz zum Erwerb des IE–Modells. Die Exploration eines solchen integrierten Verfahrens zur Konstruktion eines IE–Modells ist die zentrale Zielsetzung, die ich zusammen mit Kollegen der Universität Karlsruhe (AIFB) in obiger Schrift angehe. Grundgedanke ist die Entwicklung einer bootstrapping–Methode in der schrittweise der domänenunabhängige Kern eines IE–Systems um domänenspezifische Lexika, Extraktionsregeln und Ontologie angereichtert wird. In diesem Ansatz gehen wir davon aus, dass eine domänenspezifische Ontologie zwar von einem humanen Experten letztlich erstellt wird, dass aber das IE–System aktiv bei der Exploration des Textkorpus (oder auch des Webs) beteiligt ist, in dem es aufgrund von bereits vorhandenem Domänenwissen (das dem System ja sukzessive bereit gestellt wird) und maschineller Lernverfahren dem Ontologie–Experten hilft, den Textkorpus effizient zu "lesen", in dem das System Kandidaten von möglichen Konzepten und ihren Relationen

[10] In der Regel sind die Templatestrukturen vorgegeben. Daher ist es ein wichtiges Ziel, gültige Aussagen über das "Formulierungspotential" von Templateinstanzen durch eine exhaustive Korpusanalyse zu bestimmen, auf deren Basis dann generische Regel induziert werden können.

[11] In diesem Zusammenhang ist es wichtig zu erwähnen, dass im Rahmen einer Diplomarbeit ein von mir konzipiertes hybrides Lernverfahren implementiert wurde, dass statistische (Maximum–Entropy–Modellierung) und symbolische Lernverfahren (EBL, vgl. 1.3) zur Extraktion von binären Relationen kombiniert, vgl. [Mor02]. Die Veröffentlichung im Rahmen eines Zeitschriftenartikels ist in Vorbereitung.

vorschlägt, die vom Experten ausgewertet werden und als Grundlage zur inkrementellen Erstellung der Ontologie fungieren. Gleichzeitig wird das erkannte Domänenwissen in das IE–System integriert (z.B. dynamische Erstellung von Domänenlexikon und Extraktionsregeln), welches im nächsten Zyklus bei der Berechnung von Konzeptkandidaten verwendet wird. Das gesamte Verfahren realisiert demnach eine schrittweise, daten– und ontologiegesteuerte Adaption eines IE–System an eine Domäne.

Während der Ausarbeitung dieses Modells ist aber klar geworden, dass zwar initial der Einsatz flacher Sprachtechnologie nötig ist, um eine robuste und effiziente Vorverarbeitung des Textkorpus zu leisten. Mit zunehmender Domänenadaptivität hat sich aber auch gezeigt, dass zumindest während der Phase der Domänenadaption eine gezielte tiefere linguistische Analyse für relevante Textpassagen benötigt wird, um feinere grammatikalische Funktionen bestimmen zu können. Mit anderen Worten bedeutet dies, dass mit zunehmender Spezialisierung eines IE–System auf eine Domäne (und damit einhergehender Steigerung der Komplexität des Domänenwissens im Sinne eines Anwachsens der strukturellen Vielfalt) ebenfalls die notwendige Tiefe der linguistischen Analyse steigt. Eine Möglichkeit, dies zu realisieren, ist die oben (vgl. 1.4.1) bereits erwähnte *Integration von flacher und tiefer Sprachverarbeitung.*

Der gesamte Aspekt der Integration von flacher und tiefer Sprachverarbeitung ist ein sehr aktiver Forschungsbereich mit steigendem Interesse (man betrachte z.B. die diesjährigen Konferenzen EACL, ACL und NAACL). Es ist dabei noch offen, wie genau welche Komponenten integriert werden können oder sollen. Hilfreich wäre eine *daten–orientierte* flexible Architektur, in der es möglich wäre, ohne großen Aufwand verschiedene flache und tiefe Komponenten zu kombinieren, umso optimale Konfigurationen für verschiedene Anwendungen zu explorieren. Im BMBF–geförderten DFKI–Projekt Whiteboard haben wir eine solch flexible NL–Architektur entwickelt, die ich im nächsten Abschnitt kurz erläutern möchte.

1.5 Eine annotationsbasierte NL-Architektur

> "An Integrated Architecture for Shallow and Deep Processing", B. Crysmann, A. Frank, B. Kiefer, H. Krieger, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, and F. Xu, in: Proceedings of ACL-2002, Philadelphia, July 2002, pp. 441-447.

Anmerkungen zur Autorenschaft Der Principal Investigator von Whiteboard (Laufzeit: 2000-2002) war H. Uszkoreit. Ich war der inhaltliche und technische Projektleiter und habe wesentliche Aspekte der Architektur konzipiert und entwickelt. Da es sich bei obiger Schrift um ein Projektpapier handelt, wurden (wie es die Regel bei anwendungsbasierter Forschung ist) alle relevanten Projektmitarbeiter als Autoren aufgelistet. Ich hatte bei der Vorbereitung des Papiers die editoriale Hauptaufgabe und hielt auch den Hauptvortrag während der ACL–2002. Ich war Hauptautor der Abschnitte "Einleitung", "Architektur"

(mit Schäfer), “Flache LT” (mit Piskorski), ”lexikalische Semantik” (mit Xu) und “Diskussion”.

So, nun zum eigentlichen Thema. Zentraler Forschungsgegenstand ist die Entwicklung, Implementation und Evaluation einer neuartigen Systemarchitektur, die die Kombination verschiedener Sprachtechnologien für eine Reihe praktischer Anwendungen erlaubt. Die neue Architektur basiert auf dem Konzept eines annotierten Textes. Die verschiedenen LT-Komponenten reichern einen mit XML annotierten Text mit Meta-Information an, die ebenfalls in XML kodiert ist. Jede Komponente kann vorher zugewiesene Annotationen nutzen oder unbeachtet lassen. Die WHITEBOARD-Architektur besteht aus einer einzigen Datenstruktur, die gleichzeitig Input, Zwischenrepräsentation und Output des Systems ist (diese als XML–CHART bezeichnete Datenstruktur erweitert das im IE–Kernsystem SMES (vgl. 1.4.2) entwickelte Konzept der Text–Chart).

Die WHITEBOARD-Architektur ermöglicht die pragmatische Kombination verschiedener Verarbeitungsansätze, wobei neue Wege der Kombination flacher und tiefer Verarbeitungsmethoden aufgezeigt werden. Die zugrundeliegende wissenschaftliche Motivation des in WHITEBOARD realisierten Ansatzes kann auch als *shallow–first* bezeichnet werden, in dem Sinne, dass eine tiefe Analyse gezielt und nur auf Bedarf auf die Resultate der flachen Analyse angewendet wird. Eine andere mögliche Integration könnte als *fall–back* bezeichnet werden, in der stets eine tiefe Analyse durchgeführt wird und nur, wenn diese scheitert, eine heuristische Reparatur der identifizierten Fragmente durchgeführt wird (solch eine Strategie wird z.B. am Parc Forschungsinstitut, Palo Alto erforscht). Wir sind davon überzeugt, dass die in WHITEBOARD realisierte *shallow–first* Strategie eine größere Flexibilität für die Anwendungsmodellierung besitzt da 1) eine *fall–back* Strategie ohne großen Aufwand ebenfalls realisiert werden kann und 2) die tiefen Methoden von den flachen Ergebnisstrukturen hinsichtlich verbesserter Robustheit, Abdeckung und Effizienz profitieren.

1.6 Zusammenfassung

Das wesentliche Ziel dieses Überblickartikels war es zu zeigen, dass meine bisherigen Forschungsarbeiten entlang eines roten Fadens verliefen, nämlich der Entwicklung domänenadaptiver robuster und effizienter Sprachtechnologie. Die bisher entwickelten Methoden sind jedoch noch stark von einer externen, benutzergesteuerten Anpassung geprägt. Im gewissen Sinne handelt es sich um eine Art der passiven Domänenadaptivität, die allerdings schon durch den Einsatz einer Reihe von Lernverfahren und komplexer Strategien maschinell unterstützt wird.

Bezogen auf das Internet liegt gegenwärtig die Hauptlast bei der aktiven, qualitativen Sichtung und Auswertung des Informationspotentials beim menschlichen Benutzer. Damit dieses Potential für eine kreative Wissensbildung von Individuen und Organisationen zugänglich und nutzbar wird, liegt ein zukünftiges Forschungsinteresse in der Erforschung und Entwicklung von aktiven — auf einen Benutzer “zugehende” — aufgaben- und inhaltsorientierte, intelligente Software–Komponenten.

Danksagung

An erster Stelle möchte ich mich bei Prof. Hans Uszkoreit bedanken, mit dem ich das Vergnügen hatte, mehr als zehn Jahre wissenschaftlich zusammen arbeiten zu können und der meiner Forschungsarbeit entscheidene Impulse gab. Ich bin auch besonders Prof. Wolfgang Wahlster zu Dank verpflichtet, der meine Forschungsarbeiten ebenfalls sehr positiv beeinflusst hat. Ich möchte beiden ganz besonders für das Vertrauen danken, dass sie mir in den vielen Jahren entgegengebracht haben.

Recht herzlich möchte ich mich bei folgenden Personen für die so fruchtbare Zusammenarbeit bzw. wissenschaftlichen Diskussionen bedanken: Jaime Carbonell, Dan Flickinger, Karin Harbusch, Aravind Joshi, Martin Kay, John Nerbonne, Gertjan van Noord, Manfred Pinkal, Ivan Sag, Steffen Staab.

Einen großen Dank auch an alle Kollegen am DFKI und an der Coli! Und ein herzliches Dankeschön an die öffentlichen Geldgeber (besonders an das BMBF), die den finanziellen Rahmen für solche Forschungsarbeiten ermöglichen ("ohne Moos nix los").

Literaturverzeichnis

[AHB+93] D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. Fastus: A finite state processor for information extraction from real world text. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, 1993.

[AI97] D. Appelt and D. Israel. Building information extraction systems. Tutorial during the 5th ANLP, Washington, 1997.

[Asc01] C. Aschmonheit. Anwendung des Web Mining zum Erlernen von Kundenprofilen. Diplomarbeit in Informatik, AIFB, Universität Karlsruhe, 2001.

[BDD+97] Stephan Busemann, Thierry Declerck, Abdel Kader Diagne, Luca Dini, Judith Klein, and Sven Schmeier. Natural language dialogue service for appointment scheduling agents. In *Proc. 5th Conference on Applied Natural Language Processing*, pages 25–32, Washington, DC., 1997.

[CFS97] Ann Copestake, Dan Flickinger, and Ivan A. Sag. Minimal recursion semantics: An introduction. 1997.

[Chi00] D. Chiang. Statistical parsing with an automatically–extracted tree adjoining grammar. In *38th ACL*, Honk Kong, 2000.

[CM98] M. Califf and R. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 1998.

[CVS00] J. Chen and K. Vijay-Shanker. Automated extraction of tags from the penn treebank. In *6th International Workshop on Parsing Technologies (IWPT'2000)*, Trento, Italy, 2000.

[DKN98] Thierry Declerck, Judith Klein, and Guenter Neumann. Evaluation of the nlp components of an information extraction system for german. In *Proceedings of the first international Conference on Language Resources and Evaluation (LREC) 1998*, pages 293–297, Granada, 1998.

[DN01] T. Declerck and G. Neumann. A cascaded shallow approach to reference resolution. In *In Proceedings of EuroConference on Recent Advances in NLP, RANLP-2001*, 2001.

[Eul01] T. Euler. Informationsextraktion durch Zusammenfassung maschinell selektierter Textsegmente. Diplomarbeit in Informatik, Lehrstuhl VII, Künstliche Intelligenz, Universität Dortmund, 2001.

[GKS02] Malte Gabsdil, Alexander Koller, and Kristina Striegnitz. Natural language and inference in a computer game. Proceedings of the 19th COLING, Taipei, 2002.

[HCW01] Kevin Humphreys, Mike Calcagno, and David Weise. Reusing a statistical language model for generation. In *Proceedings of the ACL 2001 Eight European Workshop on Natural Language Generation (EWNLG)*, pages 86–91, 2001.

[Hei01] M. Heidmann. Anwendungen von linguistischen und wissensbasierten Verfahren im Information Retrieval. Diplomarbeit in informatik, AIFB, Universität Karlsruhe, 2001.

[HMS01] A. Hotho, A. Maedche, and S. Staab. Ontology-based text clustering. Proceedings of the IJCAI-2001 Workshop Text Learning: Beyond Supervision, August, Seattle, USA, 2001.

[KBB+01] Meike Klettke, Mathias Bietz, Ilvio Bruder, Andreas Heuer, Denny Priebe, Günter Neumann, Markus Becker, Jochen Bedersdorfer, Hans Uszkoreit, Alexander Maedche, Steffen Staab, and Rudi Studer. Getess — Ontologien, objektrelationale Datenbanken und Textanalyse als Bausteine einer semantischen Suchmaschine. *Datenbank-Spektrum*, 1(1):14–24, 2001.

[KBdJW02] Alistair Knott, Ian Bayard, Samson de Jager, and Nick Wright. An architecture for bilingual and bidirectional nlp. In *Proceedings of the 2002 Australasian Natural Language Processing Workshop*, pages 63–70, 2002.

[KDN98] Judith Klein, Thierry Declerck, and Guenter Neumann. Evaluation of the syntactic analysis component of an information extraction system for german. In *Proceedings of the Workshop on the Evaluation of Parsing Systems(LREC 1998)*, Granada, 1998.

[Kuh00] Jonas Kuhn. Processing optimality-theoretic syntax by interleaved chart parsing and generation. pages 360–367. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hongkong, 2000.

[Mor02] V. Morbach. A hybrid machine learning approach for information extraction. Diplomarbeit in informatik, Universität des Saarlandes, 2002.

[NBP00] G. Neumann, C. Braun, and J. Piskorski. A divide-and-conquer strategy for shallow parsing of german free texts. In *Proceedings of the 6th International Conference of Applied Natural Language*, Seattle, USA, April 2000.

[Neu94] Günter Neumann. Application of explanation-based learning for efficient processing of constraint-based grammars. In *Proceedings of the 10th IEEE Conference on Artificial Intelligence for Applications, March 1-4*, pages 208–215, San Antonio, Texas, USA, 1994.

[Neu97] Günter Neumann. An on-line learning method to speed-up natural language processing. Technical report, DFKI, Saarbrücken, 1997.

[NS02] Günter Neumann and Sven Schmeier. Shallow natural language technology and text mining. *Künstliche Intelligenz. The German Artificial Intelligence Journal*, 2002.

[PO03] Matthew Purver and Masayuki Otsuka. Incremental generation by incremental parsing: Tactical generation in dynamic syntax. pages 79–86. Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003), Association for Computational Linguistics, Budapest., 2003.

[PP03] L. Peshkin and A. Pfeffer. Bayesian information extraction network. In *Proceedings of the 18th IJCAI*, Acapulco, Mexico, August 2003.

[RY02] D. Roth and W. Yih. Probabilistic reasoning for entity and relation recognition. In *Proceedings of the 19th Coling*, Taipei, Taiwan, August 2002.

[Vol00] R. Volz. Akquisition von Ontologien mit Text–Mining–Verfahren. Diplomarbeit in Informatik, AIFB, Universität Karlsruhe, 2000.

[WCP+03] David Weir, John Carroll, Daniel Paiva, Roger Evans, Kees Van Deemter, and Anja Belz. Cogent: Controlled generation of text. http://www.cogs.susx.ac.uk/lab/nlp/cogent/ and http://www.itri.bton.ac.uk/ projects/cogent/, project Web pages at University of Sussex and University of Brighton, 2003.

[Xia99] F. Xia. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium(NLPRS-99)*, Beijing, China, 1999.

[XP00] F. Xia and M. Palmer. Comparing and integrating tree adjoining grammars. In *Proceedings of the 5th TAG+ workshop*, Paris, France, May 2000.

[YGTH00] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the 6th ANLP*, Seattle, USA, April 2000.