Korpusbasierte Techniken zum Lernen von Übersetzung spontan gesprochener Sprache

cand. inform. Klaus Ries

Diplomarbeit

Universität Karlsruhe, Fakultät für Informatik

Betreuer: Prof. A. Waibel
Finn Dag Buø

31. Juli 1994

Zusammenfassung

Bei der maschinelle Übersetzung spontaner gesprochener Sprache hat sich die Verwendung traditioneller symbolischer Methoden als ausgesprochen schwierig erwiesen. In dieser Arbeit wird daher versucht, auf rein statistischem Wege aus übersetzten Texten ein Übersetzungssystem vollautomatisch abzuleiten. Dazu sollen zunächst aus manuellen Übersetzungen automatisch Übersetzungsbeispiele erstellt werden, mit denen ein Übersetzungssystem gelernt werden soll.
Dazu werden zunächst neue statistische Methoden zur einsprachigen, unüberwachten, hierarchischen Corpusanalyse entwickelt, die sowohl für ein komplexitätstheoretischen Maß als auch für die Perplexität gute Resultate aufweisen und als Sprachmodell in Konkurrenz zu n-gramm Modellen stehen. Die mit diesen Methoden erzeugten Wortklassifikationen und Verkettungen von Worten wurden auch zur Unterstützung der Identifikation von Sprechakten benutzt.
Das Übersetzungssystem sollte sich in zwei Bestandteile gliedern, ein Verfahren zur Beispielgenerierung durch eine Wortzuordnung und ein Verfahren zum Training einer Transferkomponente. Das Verfahren zur Wortzuordnung wurde vor allem um eine besondere Berücksichtigung von Floskeln und Wortfeldern erweitert. Das Verfahren zur Wortzuordnung war auf diesem schwierigen Corpus allerdings nicht erfolgreich. Es werden daher Transfermethoden vorgestellt, die ohne eine Wortzuordnung auskommen und es werden Methoden angegeben, die die Erstellung von solchen Zuordnungen unter Einbeziehung weiterer Wissensquellen auf strukturellem Wege erlauben.

Vorwort

Die Erstellung dieser Arbeit konnte ohne die freundliche Unterstützung vieler Menschen kaum gelingen. Zuerst seien hier meine Eltern genannt, nicht nur aber auch weil sie es mir finanziell ermöglichten, diese Arbeit zu großen Teilen an einem eigenen Rechner zu entwickeln und zu schreiben. Die wissenschaftliche Betreuung durch Prof. A. Waibel und Finn Dag Buø haben mir an vielen Stellen geholfen und zur Revision von Ansätzen angeregt. Insbesondere möchte ich mich für das beharrliche Schubsen bedanken, die verwendeten Methoden immer an praktischen Beispielen zu bewerten.
Ein großer Dank geht auch an Thomas Polzin, der Teile dieser Arbeit Korrektur gelesen hat und durch seine Anmerkungen zu einer Verbesserung der Präsentation und Fokussierung beigetragen hat. Unter den Mitarbeitern am Institut möchte ich keine Unterscheidungen mehr machen – obwohl sie sich eher mit der Spracherkennung beschäftigen gab es reichlich Stoff für Diskussionen und immer wieder Hilfestellung.
Ein großer Dank geht auch an die Friedrich Naumann Stiftung für ihre finanzielle und ideele Unterstützung. Viele Menschen, die ich dort kennengelernt habe, haben mir in dieser Zeit mehr geholfen, als sie es selber wissen. Ebenso muß ich mich bei meinen Freunden bedanken, die mich während dieser Zeit ertragen haben.
Es bleibt ein lachendes Gesicht zurück, weil ich endlich fertig bin, und es bleibt ein weinendes Gesicht zurück, weil es vorbei ist.

Verfügbarkeit der erstellten Software

Die im Rahmen dieser Diplomarbeit erstellte Software und verwendeten Corpora stehen zunächst institutsintern zur Verfügung und sind mit einer entsprechenden Dokumentation in Englisch versehen. Als Voraussetzung ist in der Regel lediglich ein C-Compiler und die Interpretersprache PERL auf dem Betriebssystem UNIX erforderlich. Ab Ende 1994 sollen die verwendeten Programme ohne die Corpora nach dem Willen des Autors auch extern verfügbar gemacht werden und auf dem Server *i13d11.ira.uka.de* für eine Zugriff über *anonymous ftp* zur Verfügung stehen. Für weitere Rückfragen bezüglich der Verfügbarkeit kann man sich an den Autor unter der Adresse `ries@informatik.uni-freiburg.de` oder an den Betreuer Finn Dag Buø `finndag@ira.uka.de` wenden.

Die vorliegende Arbeit wurde als Diplomarbeit angefertigt an der

Universität Karlsruhe (Technische Hochschule)
Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Postfach 6980
D-76128 Karlsruhe

Die Betreuer der Arbeit waren Prof. A. Waibel und Dipl.-Inform. Finn Dag Buø.

Das Studium wurde auch während der Dauer dieser Arbeit gefördert durch die

Friedrich-Naumann-Stiftung
Wissenschaftliche Dienste und Begabtenförderung (WDB)
Königswintererstr. 407
D-53639 Königswinter

Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbstständig erstellt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu habe.

Klaus Ries

Karlsruhe, den 31.07.1994

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einführung

Das Ziel des Janus -Systems ist die Verwirklichung der Vision eines Computers, der gesprochene Sprache von einer in eine andere Sprache übersetzt und gesprochen wieder ausgibt. Zusätzlich zur reinen Sprachinformation sollen in Zukunft auch die Lippenbewegungen und die Gestik analysiert und mitübertragen werden. Die Anwendungen eines solchen Systemes in einer sich stetig internationalisierenden Welt sind vielfältig. Wirtschaftlich interessant sind aber auch bereits Teilkomponenten des Systems wie z.B. die reine Spracherkennung und die Lokalisierung von Gesichtern im Raum. Diese Systeme können z.B. für die Erstellung moderner Benutzungsschnittstellen für Computer verwendet werden, womit sich weitere, angeschlossene Projekte befassen.

Diese Arbeit wird sich ausschließlich mit der Sprachverarbeitung und -übersetzung beschäftigen. In der Abbildung 1.1 kann man die derzeitige Konzeption des Janus -Systems erkennen [WAWB+94]. Nach der Spracherkennung sind drei unterschiedliche Parser[1] angeschlossen, die auf unterschiedlichen Konzepten beruhen und daher eine unterschiedliche Analysequalität bieten (siehe Abbildung 1.2). Das Ergebnis dieser Parser ist eine abstrakte, semantische Repräsentation der Eingabe, die Interlingua (siehe Abbildung 1.3). Die Interlingua kann mit Sprachgeneratoren in viele Sprachen übersetzt werden. Während das Problem der Generierung aus einer Interlingua in eine neue Sprache eine vergleichsweise einfache Aufgabe darstellt, ist die Analyse der Daten, die vom Spracherkenner erzeugt werden, ausgesprochen schwer. Dies hat mehrere Ursachen

- der Spracherkenner macht Fehler
- das Problem des Parsings von Schriftsprache ist nicht endgültig gelöst
- die Grammatik gesprochener Sprache ist bei weitem nicht so gut untersucht wie die von Schriftsprache
- gesprochene Sprache zeichnet sich durch eine sehr hohe Variabilität aus und ist prinzipiell nur schwer zu fassen
- Menschen machen Fehler während der Produktion von Sprache
- der Mensch kommuniziert auch nicht-verbal (Intonation, Gestus) und diese Information kann entscheidende Information über den Sinn eines Satzes geben

[1] Die Analyse eines Textes mit einer spezifischen Technik (also nicht durch Introspektion) soll in dieser Arbeit als *Parsing* bezeichnet werden, ein Programm zum *Parsing* ist daher ein *Parser*. Von einigen deutschen Autoren wird der Begriff Zerteilung bevorzugt, der allerdings voraussetzt, daß die Analyse der Bestandteile zu einer Gesamtanalyse zusammengebaut wird, was nicht notwendig ist.

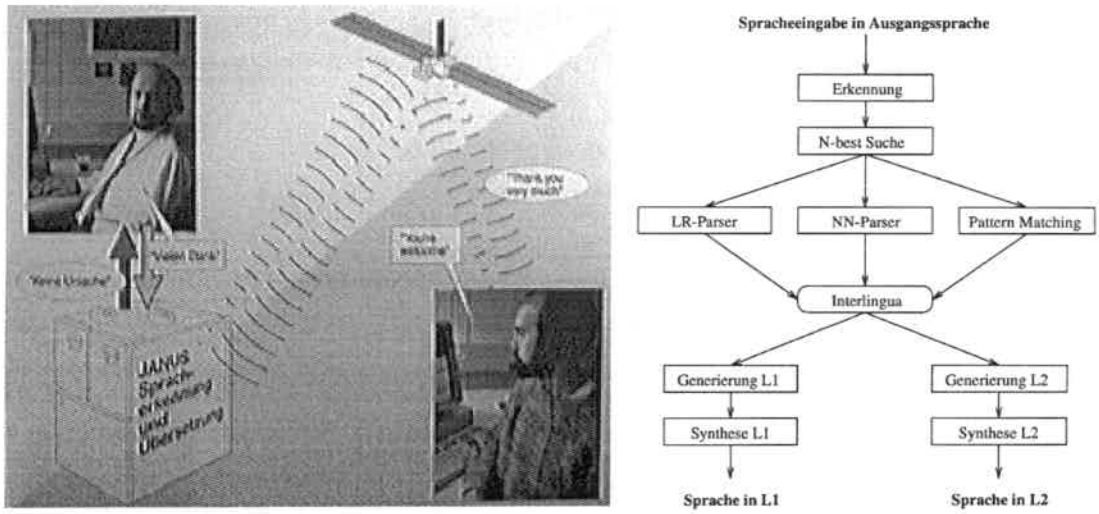


Abbildung 1.1: **Vision und Realisierung des Janus -Systems:** Die Vision des Janus -Systems ist die Herstellung der Kommunikationsfähigkeit zwischen verschiedensprachigen Partnern. Ein Spracherkenner und die nachgeschaltete Suche wandeln dazu das Sprachsignal in Zeichen um, verschiedene Parser versuchen diese Eingabe in eine abstrakte, sprachunabhängige Repräsentation der Äußerung (Interlingua) zu übersetzen, die von Generierungsmodulen in eine Schriftform der Ausgabesprache umgesetzt wird. Ein Sprachsynthesemodule macht dann die generierte Sprache wieder hörbar.

Die verwendeten Parsingtechniken sind

Skipping G-LR Parser Eine Implementierung eines generalisierten LR-Parsers, der Eingabesymbole überspringt, die er nicht interpretieren kann. Ein eingeschränkter Suchprozeß versucht, die Anzahl der übersprungenen Symbole so gering wie möglich zu halten. Die verwendete Grammatik ist semantischer Natur, Versuche mit klassischen syntaktisch orientierten Grammatiken haben nicht zum Erfolg geführt.

Pattern Matching Auf Basis der rekursiven ATN's (*augumented transition networks*) wurde ein Pattern-Matching Algorithmus entwickelt, der ebenfalls mit einem Suchprozess versucht, die durch ihn abgedeckten Bereiche des Textes zu maximieren.

neuronale Parser Der von [Jai91a] gebaute neuronale Parser wurde um die Ausgabe von strukturierten Repräsentationen erweitert [BPW94]. Der Parser zerhackt seine Eingabe mit einem Netz in mehrere Teile (auch *chunks* genannt) und klassifiziert diese einzeln. In diesen Parser wurde erfolgreich prosodische Information integriert und der Parser ist sehr unempfindlich gegen fehlerhafte Eingaben.

Die Idee für diese Diplomarbeit bestand darin, die statistische Analyse zweisprachiger Texte (im folgenden auch Corpora genannt) und von Sätzen mit vorgegebenen Interlinguastrukturen für die Übersetzung im Janus -System zu benutzen. Dazu sollten die folgenden Methoden verwendet werden:

Alignment Erstellung eines Programmes, das die Bestandteile von zweisprachigen Texten einander gegenüberstellt

Verbesserung des Alignment Verbesserung dieses Programmes durch Ansätze zur Wortklassifikation und Sequenzensuche

Transfer Erstellung eines Transfermodules, das aus den in den ersten zwei Schritten erstellten Übersetzungsbeispielen einen Übersetzer generiert

Bei der Erstellung des Übersetzungsmodules war daran gedacht, möglicherweise nur Interlingua zu erzeugen, da auf diese Weise u.a. Probleme mit den abtrennbaren Prefixen des Deutschen vermieden würden (z.B. ab-sagen). Wie im folgenden gezeigt wird, stellte das Alignment eine ganz entscheidende Hürde bei dieser Arbeit dar, und daher wurde das Arbeitsprogramm abgeändert. Während die Ergebnisse beim Alignment auch durch die eingefügten Verbesserungen nicht ausreichten, um eine Transferkomponente zu traineren, waren die Algorithmen zur Wortklassifikation und Sequenzensuche relativ effizient und genau.

1.1 Maschinelle Übersetzung

Das Problem der Übersetzung von einer Sprache in eine andere ist ein lang und gut untersuchtes Thema. Was aber ist eigentlich eine Übersetzung, welchen Kriterien muß oder kann sie genügen? Hierzu schreibt [Lew90, Übersetzung]

inhaltliche Äquivalenz Der Sachverhalt bzw. der Sinn einer Äußerung wird inhaltlich korrekt wiedergegeben.

konnotative Äquivalenz Die Stilschicht, die soziolektale und geographische Dimension wird erhalten.

textnormative Äquivalenz die sprachlichen Gebrauchsnormen werden erhalten (z.B. Novellenstil).

pragmatische und kommunikative Äquivalenz Der Empfänger der Übersetzung soll die Übersetzung in der vom Sender intendierten Weise verstehen.

Die Entwicklung der maschinellen Übersetzung läßt sich in folgende Phasen gliedern [Lew90, automatische Übesetzung]

Computerzentrierte Phase Auf Rechnern wurden durch einfache, rein lexikalische Prozeduren der Text im wesentlichen Wort für Wort übersetzt.

Linguistische Phase Durch die Kritik [ALP66] angeregt wurden die einfachen linguistischen Annahmen durch komplexere ersetzt und eine abstrakte Repräsentationsebene von Sprache angenommen. Dazu wurden grundlegende Arbeiten im Bereich der Syntax geleistet.

KI-Modelle Die durch vor allem syntaxorientierte Analysen erreichbare Genauigkeit ist durch das Problem der syntaktisch nicht auflösbaren Mehrdeutigkeit stark eingeschränkt. Durch rein syntaktische Transformationen kann darüberhinaus der Bedeutungsgehalt des Satzes massiv verändert werden. Die KI-Phase ist daher dadurch gekennzeichnet, daß man in einer ersten Analysephase die Bedeutung oder den kommunikativen Aspekte eines Satzes extrahiert und dann in einer zweiten Phase versucht, dies in der zweiten Sprache auszudrücken.

In der ersten Phase der KI-Modelle wurde versucht, die Bedeutung der Sprache möglichst genau zu analysieren, um *high-quality* Übersetzungen zu erreichen. Aus pragmatischen Überlegungen wurden dann in einer zweiten Phase dazu übergegangen, *good-enough* Übersetzungen zu erzeugen. Dabei wird das ambitiöse Ziel, die Sprache als Ganzes zu verstehen, durch das Ziel ersetzt, einen in einer bestimmten Sprachsituation (Domäne) relevanten bzw. zwingend benötigten Bestandteil zu modellieren. In der oben aufgeführten Klassifikation von Übersetzungsgüte wird

	G-LR	Pattern Matching	Neuronal
Genauigkeit	Gut	Ungeklärt	Mittel
Zurückweisung von Eingaben	Oft	Manchmal	Selten
Robustheit	Mittel	Mittel	Hoch
Geschwindigkeit	Sehr langsam	Langsam	Mittel
Entwicklungszeit	Sehr hoch	Hoch	Mittel
außersprachliche und prosodische Information	Nein	Nein	Ja

Abbildung 1.2: **Parser des Janus -Systems im qualitativen Vergleich:** Die Parser des Janus -Systems beruhen auf unterschiedlichen Prinzipien und diese Prinzipen implizieren die Stärken und Schwächen der einzelnen Parser. Diese Übersicht kann nur eine Tendenz wiedergeben und durch Änderungen an den Implementierungen schnell veralten. Das Kriterium der *Genauigkeit* steht für die Korrektheit der Analyse. Die Rate der *Zurückweisung von Eingaben* gibt an, wie oft der Parser eine Eingabe akzeptiert und eine Ausgabe generiert. Die *Robustheit* bezeichnet die Toleranz des Parsers gegenüber Eingaben, die nicht im modellierten Sprachfragment vorgesehen sind. Die *Geschwindigkeit* ist die Zeit für eine Analyse, die *Entwicklungszeit* bezeichnet die Arbeitszeit von Grammatikentwicklern zur Übertragung des Parsers auf eine neue Sprache oder neue Anwendung. Die letzte Zeile gibt die Fähigkeit an, *außersprachliche und prosodische Information* beim Parsing zu integrieren.

```
A-SPEECH-ACT     *PROPOSE-MEETING
D-SPEECH-ACT     *PROPOSE-MEETING
SENTENCE-TYPE    *STATE
WHO              FRAME           *WE
WHEN             FRAME           *SPECIAL-TIME
                 SPECIFIER       *MULTIPLE* DEFINITE NEXT 2
                 NAME            WEEK
WHAT             FRAME           *MEETING
HOW-LONG         FRAME           *LENGTH
                 SPECIFIER       APPROXIMATE
                 HOUR            2
```

Abbildung 1.3: **Interlingua des Janus -Systems:** Das Janus -System benutzt eine auf die Domäne "Terminabsprachen" hin modellierte semantisch/pragmatische Zwischensprache (Interlingua). Die textuelle Repräsentation wird auch Interlingua Text (ILT) genannt. In der Interlingua spielen vor allem Zeitspezifikationen und Klassifikationen der Sätze nach ihrer pragmatischen Information, z.B. *suggesting*, eine Rolle. Der analysierte Satz ist *Maybe we can get together and discuss the planning for say two hours in the next couple weeks*

also versucht, die inhaltliche Äquivalenz zu erreichen. In der vorliegende Domäne von Terminabsprachedialogen ist das vorgegebene Ziel allerdings die Herstellung der Kommunikationsfähigkeit zwischen den Partnern, was eine Abschwächung der Bedingung an die inhaltliche Wiedergabegüte beinhaltet[2], aber höhere Anforderungen an die exakte Wiedergabe der pragmatischen Information stellt[3].

Aus verschiedenen weiter unten zu besprechenden Gründen hat sich selbst die Erzeugung von *good-enough* Übersetzungen als ausgesprochen schwierig erwiesen und es wurden in den letzen Jahren eine Reihe von Ansätzen entwickelt, die das (statistische) Wissen aus (zweisprachigen) Texten für die Übersetzung verfügbar machen sollen [Iid93].

1.1.1 KI-Modelle

Selbst für das oben beschriebene Ziel der Erfassung der relevanten Information für eine *good-enough* Übersetzung ist ein sehr hoher Aufwand zur Erstellung eines Systems erforderlich. Dieser Abschnitt soll im einzelnen klären, wie sich dieser Aufwand auf das System verteilt und welche Probleme diese Übersetzungssysteme haben.

Das KI-Modell postuliert zunächst eine allgemeine oder an der zu lösenden Aufgabe orientierte inhaltliche Beschreibung der möglichen Sätze der Gesamtsprache bzw. der zu behandelnden Teilsprache (Interlingua). Diese Modellierung geschieht in der Regel auf semantischer Basis, oft aber auch auf pragmatischer Ebene. Eine Grundidee der Übersetzung mit einer Interlingua ist die Sprachunabhängigkeit des resultierenden Systems, da für jede Sprache in einem Übersetzungssystem nur eine Analyse nach Interlingua und eine Synthese aus Interlingua benötigt wird. Bereits das Postulat einer für alle Sprachen gültigen Interlingua bzw. der Vorgang der Erstellung derselben sind sehr problematisch. Neben dem Problem der Erstellung einer allgemeinen Ontologie, für die eine praktikable Lösung in [HN92] besprochen wird, ergeben sich prinzipielle Probleme bei der Erstellung eines Übersetzers mit einer konstruierten und abstrakten Zwischensprache:

Introspektion begrenzt möglich Die Erstellung der Interlingua ist ein schöpferischer Prozeß des Erstellers der Grammatik, der zunächst vor dem Beginn der Arbeit steht und zunächst in seiner Konsequenz nicht zu überprüfen ist.

Sprachabhängigkeit vs. -unabhängigkeit der Interlingua Der Phase der Erstellung der Interlingua erfolgt in aller Regel eine Phase, in der die Interlingua an einem konkreten Sprachpaar evaluiert wird. In dieser Phase werden die nach Meinung der Ersteller der Grammatik zusätzlich benötigten Informationen eingefügt. Dabei erhöht sich der Bias auf das verwendete Sprachpaar, den die Ersteller schon während der Konzeption erzeugt haben: Die Einheiten der Interlingua müssen mit einer Bedeutung verbunden werden und diese Bedeutung muß in einer Referenzsprache erfolgen – in aller Regel wird dazu eine Mischung der Semantik der Zielsprachen benutzt. In [HN92] wird beschrieben, wie man trotz sprachabhängiger Formulierung neue Sprachen schrittweise in das System integrieren kann – die Komplexität der dort beschriebenen Lösung macht jedoch die entstehenden Probleme deutlich. Insbesondere erschwert die inkrementelle Entwicklung der Interlingua die Erstellung der Parser.

[2] Der genaue Grund, aus dem ein Termin nicht zustandekommt, kann für die Herstellung der Kommunikationsfähigkeit nicht so entscheidend sein.

[3] Der Grad der Zustimmung zu einem Terminvorschlag kann auf komplexe Weise in der Äußerung kodiert sein oder gar nur in der Intonation existieren, kann aber entscheidend sein für eine gelungene Interaktion.

Verlust von sprachpaarabhängigen Transferwissen Die Erstellung der Interlingua versucht, die Sprachpaarabhängigkeit im Übersetzungsprozeß zu eliminieren. Gerade die speziellen Eigenschaften von Sprachpaaren machen jedoch gute Übersetzungen möglich, ohne eine vollständige Analyse zu benötigen. Dies trifft zum Beispiel auf das Problem der Wortselektion in der Zielsprache zu, das sprachpaarabhängig in der Regel einfacher zu lösen ist und auch auf das Problem der Erhaltung anaphorischer Referenz.

Verlust der Satzstruktur Die Übersetzung der Ausgangssprache in eine abstrakte Repräsentation führt unmittelbar zu einem Verlust der Struktur des Satzes. In der Struktur eines Satzes können aber wesentliche Informationen kodiert sein, die vor allem pragmatischer Natur sind. Dazu gehören bei gesprochener Sprache u.a. Intonationsinformation, Korrekturen und die Reihenfolge der Konstituenten im Satz. Der Transfer solcher Information setzt also eine Analyse des Satzes, der diesen Aspekt berücksichtigt, mit ein. Ein korrelierter Transfer von Gestik kann nach der Generierung einer Interlingua nicht mehr erfolgen.

Trotz aller dieser Probleme wird dieser Ansatz heute in den meisten Projekten verfolgt. Im Janus Projekt hat es sich gezeigt, daß die Synthese der Zielsprache ein relativ gut zu lösendes Problem ist, während die Analyse der spontanen Sprache recht schwierig ist. In internen Experimenten bei der Janus Evaluation wurde das Phänom beobachtet, daß oft korrekte Übersetzungen aus einer falschen Interlingua erzeugt wurden.

Eine große Hürde im Janus Projekt ebenso wie in anderen Projekten stellt derzeit die Erstellung der Parser dar, die durch einen hohen Aufwand an Entwicklungszeit für Grammatiken beim LR-Parser bzw. der Erzeugung von Trainingsdaten und der Adaption beim neuronalen Parser gekennzeichnet sind. Eine Beurteilung des Pattern-Matching Ansatzes kann zum Zeitpunkt der Erstellung dieser Arbeit noch nicht erfolgen, da hiermit noch keine Übersetzungen gemacht wurden. Die Probleme bei der Erstellung der Parser wurden bereits oben angeschnitten. Die spezifischen Probleme mit der gesprochenen Sprache werden in Abschnitt 1.2 besprochen. Die Kritik, die von [Iid93] vorgebracht wurde, bezieht sich von daher auch vor allem auf das Ungleichgewicht zwischen den Komponenten *Vorverarbeitung*, *Analyse* und *Generierung*, bei denen die *Analyse* eine unverhältnismäßig hohe Menge an Arbeit zu leisten hat. Dieses Problem wird vor allem für den von Hand entwickelten Parser durch die bereits im Projekt zu beobachtenden Änderungen an der Interlingua, die durch die oben angeschnittenen Probleme in gewissem Umfang prediziert werden, erhöht.

1.1.2 Statistische Methoden

Die Anwendung von statistischen Methoden in der maschinellen Übersetzung ist mit der Hoffnung verknüpft, in kurzer Zeit brauchbare Übersetzungssysteme für eingeschränkte Domäne zu erstellen. Zum einen kann die schnelle Erstellung von Übersetzungssystemen aus Parallelcorpora das sprachpaarabhängige Transferwissen ausschöpfen und zum anderen besteht die Hoffnung, daß ein so erstelltes System eine hohe Robustheit gegenüber den Eingaben spontan gesprochener Sprache besitzt. Zusätzlich wird durch einen lernenden Ansatz der Mangel in der Formalisierung der Grammatik gesprochener Sprache (siehe Abschnitt 1.2) umgangen und eine Integration weiterer Wissensquellen wie Prosodie und Gestik erleichtert.

Unter den statistischen Methoden haben sich folgenende "Schulen" ausgebildet:

Alignment Die Aufgabe des Alignments ist es, Sätze oder Worte in einem zweisprachigen Corpus ihren Übersetzungen zuzuordnen [BCP+90] [BPPM91] [BPP+] [BPPM93a] [KR93] [CDG+93b] [GC93] [CDG93a].

Example Based (EBMT) und Transfer Driven (TDMT) Übersetzung In einer komplexen Übersetzungssituation soll die Auswahl von Worten und Konstruktionen anhand eines Beispieles (d.h. im wesentlichen anhand eines beschränkten Kontextes) geleistet werden [FI92] [SI92] [Iid93].

Holistische Übersetzung Mithilfe von Kompressionsverfahren wird eine Repräsentation des zu übersetzenden Satzes erstellt und diese wird mit einem Lernverfahren in eine Repräsentation der Zielsprache übertragen [Cha90] [Chr91].

Weder das Alignment noch das Example Based Transfer Verfahren sind für sich genommen Methoden, mit denen sich ein Übersetzungssystem ohne oder mit geringer menschlicher Intervention erstellen ließe. Während bei den Arbeiten zum Alignment meist eine Wort-für-Wort Übersetzung mit einer Umordnung der Worte nach statistischen Gesichtspunkten erfolgt, gehen die Autoren von EBMT und TDMT bereits von einer erstellten Basis von Beispielen auf Wort bzw. Phrasenebene aus. Die holistische Übersetzung hingegen benötigt in der Regel lediglich ein Alignment auf Satzebene, daß sehr viel leichter und sicherer automatisch erstellen läßt als ein Alignment auf Wortebene[4]. Eine der Ideen, die dieser Arbeit zugrunde lagen, war die Verknüpfung des Alignment mit einem besseren Transfersystem, z.B. dem EBMT oder aber einem anderem Lernsystem (z.B. dem Error Driven Transformation Learning [Bri92][Bri93]). Dazu sollten Beispiele aus den durch das Alignment gewonnen Daten automatisch generiert werden. Diese Verfahren werden im Kapitel 4 vertieft.

Zusätzlich zu den vorgestellten Methoden können solche statistischen Methoden verwendet werden, die eine Analyse eines einsprachigen Corpus ermöglichen. Solche Methoden können mit vergleichsweise hoher Sicherheit und Präzision angewendet werden. In dieser Arbeit werden ausschließlich unüberwachte Verfahren verwendet, die Wort- und Phrasenklassifikation sowie eine Verkettung von Worten bzw. Phrasen zu längeren Einheiten ermöglicht. Eine Vertiefung erfolgt im Kapitel 3.

1.2 Besonderheiten der zugrundeliegenden Domäne

Janus ist ein System, das unmittelbar gesprochene Sprache verarbeitet. Während die geschriebene Sprache relativ gut untersucht ist, stellt die Analyse von gesprochener Sprache ein schwerwiegendes Problem für die derzeit vorherrschende Richtung in der Linguistik dar. So argumentieren [Pil90][Hal90], daß die begriffliche Abstraktion des Satzes als syntaktische Einheit für die Analyse gesprochener Sprache nicht sehr nützlich ist. Das primäre von [Pil90] gegen eine syntaktische Analyse angeführte Argument ist das Vorhandensein von Doppelbindungen in einem Satz wie

I hate | sitting around [here] because I'm in a bad mood | I want to go home

Im vorliegenden Beispiel müßte man in einer traditionellen Analyse eine Entscheidung treffe, ob *because I'm in a bad mood* zum vorangehenden oder zum nachfolgenden Hauptsatz gehört. Tatsächlich kann man aber keine eindeutige Zuordnung

[4] Das Alignment auf Satzebene war für diese Arbeit bereits gegeben und mußte nicht mehr erstellt werden.

treffen, der Nebensatz ist mit beiden verknüpft. Damit verstößt das obige sogenannte Flechtbandsyntagma gegen das Linearitätsaxiom der Sprachwissenschaft, wonach eine Bindung wie oben zwar mehrdeutig sein dürfte, es aber immer eine eindeutige Auflösung gibt. [Pil90] zeigt ebenso, daß die Analyse altenglischer Texte mit klassischen Methoden auf ähnliche Probleme stößt wie die Analyse gesprochener Sprache. Weitere, spezifisch mündliche Syntagmen sind

Topic Movement Das Thema des Satzes wird an den Anfang oder das Ende des Satzes gestellt (Brenda Jeff never fascinated her).

Wiederholungen Durch eine Wiederholung findet eine Präzisierung und Zuspitzung eines Begriffes statt (he didn't say he couldn't say a word).

Schrittweise Information Statt länglicher Spezifikationen (z.B. durch Adjektive) wird eine schrittweise Spezifikation vorgenommen (the people who came in were Palatinate Germans who had been recruited activly recruited by the British Crown).

Einbettung In mündlicher Sprache findet nach dem Ende einer Einbettung oft eine Wiederholung, Fortsetzung oder Variation des Syntagmas vor Beginn der Einbettung statt (*lipsmack um* saturday the fourth wouldn't be too bad and *uh* neither would *uh kurze Pause* sunday the twenty eighth *kurze Pause uh* sometime in the afternoon *kurze Pause* but if *em* saturday seems to be most convenient then *uh kurze Pause* that it is *click*).

Nach [Hal90] sind alle diese Syntagmen nicht durch den traditionellen Syntaxbegriff zu erschließen und sind dennoch fester Teil der Kommunikation. [Hal90] schließt sich [Mul89] an, der eine Trennung in eine syntaktische und eine parasyntaktische oder sententielle Ebene vorgeschlagen hat. In dieser Analyse bewegt sich die geschriebene Sprache fast ausschließlich im syntaktischen Bereich, während für die Analyse der gesprochenen Sprache der Kommunikationsaspekt zunehmend wichtiger wird und sich in festen Syntagmen ausdrückt. Die zweifache Anbindung oben kann daher als eine sententielle Anbindung analyisert werden, die nicht mehr den Regeln der Syntax unterliegt. [Pil90] argumentiert, daß diese Unterschiedlichkeit nicht zufällig ist, sondern in der Unterschiedlichlichkeit der Medien "geschriebener Text" vs. "gesprochene Sprache" begründet liegt. Die besprochenen Syntagmen liegen ausschließlich in der Kommunikationsstruktur der Sprache und sind in mehreren Sprachen zu beobachten. Im Rahmen eines Übersetzungssystems für gesprochene Sprache ist also an einen direkten Transfer solcher Syntagmen zu denken. Man kann also bei gesprochener Sprache mit einer guten Performanz eines Ansatzes rechnen, der Templates direkt transferiert.

Zusätzlich zu den hier besprochenen Phänomenen gesprochener Sprache kommen bei spontan gesprochener Sprache weitere Phänomene, die durch die Performanz des einzelnen Sprecher bei der Erzeugung von Sprache bedingt sind, wie z.B. Stottern, Korrekturen und Planungsfehler. In den Ausgaben des Erkenners finden sich zudem in engem Rahmen auch prosodische Information z.B. die Sprechpausen in zwei verschiedenen Längen. Weitere prosodische Information kann teilweise aus dem Erkennungsprozeß extrahiert werden und wird derzeit nur vom neuronalen Parser benutzt. In den Transkriptionen von Texten, wie sie in dieser Arbeit verwendet wurden, finden sich neben den Pausen keine weiteren prosodischen Markierungen, sodaß diese hier nicht verwendet werden konnten.

1.3 Gliederung der Arbeit

In der **Einführung** wird der Kontext und die Zielrichtung dieser Arbeit beschrieben. Dazu werden insbesondere das Janus -System und die derzeit dort verwendeten

Ansätze zur Übersetzung beschrieben und in einen größeren Rahmen gestellt. Als Grundlage dieser Arbeit werden die statistischen Methoden zur Übersetzung, die in einem späteren Kapitel vertieft werden, angesprochen. Schließlich werden die Besonderheiten der spontan gesprochenen Sprache und deren derzeitiger Untersuchungsstand in der Linguistik besprochen.

Im Kapitel **Technische Grundlagen** werden für diese Arbeit relevante bzw. realisierte und modifizierte Verfahren und Maße aus der Statistik, Numerik und Informationstheorie dargestellt und auf die entsprechendenden Implementierungen referiert, die erstellt bzw. benutzt wurden. Zu den implementierten Verfahren zählen dabei insbesondere ein leistungsfähiges Clusterverfahren und Erweiterungen eines Simulators für neuronale Netze.

Das Kapitel zur **Unüberwachten hierarchischen Analyse einsprachiger Corpora** stellt Verfahren zur Klassifikation von Worten bzw. Wortfolgen und zur Identifikation von zusammengehörigen Folgen von Worten vor, die zwar Vorläufer haben, aber in dieser Form nicht bekannt sind. Das Kapitel führt ebenfalls ein komplexitätstheoretisches Maß für die Messung der Güte der abgeleiteten Modelle ein und stellt einen Vergleich auf der Basis der Perplexität zu n-gram Modellen her und zeit, daß dieses Modell die Daten deutlich besser erklären kann als n-gramme. Weiterhin werden Experimente zur Sprechaktidentifikation vorgestellt.

Das Kapitel zur **Übersetzung mit statistischen Methoden** stellt dar, welche Methoden zur Erstellung eines statistischen Übersetzungssystems aus einem zweisprachigen Corpus bekannt sind und stellt dar, welche Probleme sich mit den jeweiligen Methoden ergeben. Insbesondere wird vorgestellt, auf welche Weise die Generierung von Übersetzungsbeispielen durch die Angabe eines Alignments in Experimenten angegangen wurde. Dazu wurden u.a. auch Verfahren zur einsprachigen Corpusanalyse benutzt. Obwohl die Verfahren zur Generierung von Übersetzungsbeispielen nicht so gut waren, um auf diese Weise eine Transferkomponente trainieren zu können, werden in dem Kapitel unterschiedliche Verfahren dafür dargestellt. Eine Darstellung von Methoden zur Übersetzung ohne ein Alignment rundet das Kapitel ab.

Schließlich wird eine **Schlußfolgerung** gezogen, die zum einen die erreichten und die offengebliebenen Ziele beschreibt und zum anderen Vorschläge für weitere Arbeiten auf diesem Gebiet unterbreitet.

Kapitel 2

Technische Grundlagen

In diesem Kapitel sollen technische Grundlagen besprochen werden, die allgemeineren Character haben, aber wichtige Voraussetzungen sind, um gute Resultate bei den technischen Problemstellungen in dieser Arbeit zu bekommen. Um dieses Kapitel so kurz wie möglich zu gestalten, sind viele allgemein bekannte Resultate nur zitiert und nur die für den Erfolg der vorliegenden Arbeit erforderlichen Resultate und Methoden vertieft. Zusätzlich werden an einigen Stellen Resultate und Techniken präsentiert, die in dieser Form nicht in der Literatur vorgefunden wurden.

2.1 Das Clusterproblem

Die Darstellung der allgemeinen Grundlagen zu Clusteranalyseverfahren kann in einem Kapitel eine Diplomarbeit kaum besprochen werden. Nach [SL77, Seite 11] kann man Clusteranalyse wie folgt definieren

> Formal gesehen, besteht das Problem in all diesen Situationen darin, meist sehr viele Objekte, Einheiten oder Elemente in kleinere und homogene Gruppen, Klassen oder *Cluster* (eng. = Haufen, Traube) aufzuteilen. Die zu gruppierenden Elemente werden durch zahlreiche Eigenschaften, Merkmale oder Variablen charakterisiert. Die auf diese Problemstellung bezogenen mathematisch-statistischen und heuristischen Verfahren der multivariaten Datenanalyse werden im folgenden zusammenfassend als *Clusteranalyse* bezeichnet.

In dieser Arbeit werden unter diesem Begriff nur solche Verfahren verstanden, die eine Einteilung einer Menge von Vektoren reeler Zahlen in Gruppen ohne weitere Vorabinformation bezüglich eines bestimmten Optimalitätskriteriums ermöglichen. Das in dieser kurzen Einführung ausschließlich betrachtete Optimalitätskriterium ist das Varianzkriterium bzw. das Fuzzy Funktional.

Eine praxisorientierte Einführung zu klassischen Cluster Algorithmen ist in [SL77] gegeben. In der Praxis haben sich bei den hier besprochenen Problemgrößen direkte Verfahren bewährt. Eine Verallgemeinerung des Varianzkriteriums führt zu Fuzzy-Algorithmen [Bez73][CDB86][Bez92][Yan93]. Weitere verwandte Clusterverfahren sind aus dem Bereich der neuronalen Netze bekannt, zu ihnen zählen neben den selbsorganisierenden Karten nach Kohonen auch die ART Architektur von Grossberg [Bra91][Roj93]. Die zu Beginn dieser Arbeit zur Clusteranalyse verwendeten selbstorganisierenden Karten nach Kohonen konnten wegen der Ungenauigkeit der Klassifikation und des hohen Rechenaufwandes nicht weiterverwendet werden.

2.1.1 Varianzkriterium und Fuzzy Funktional

Wie oben erwähnt, sind nicht beliebige Klassenbildungen von Interesse sondern genau solche, die bezüglich eines genau definierten Qualitätsmaßes eine optimale Klasseneinteilung bedingen. Dazu wird ein Modell der Daten erstellt, bei dem zu jeder Klasse i ein sogennanter Clustermittelpunkt V_i und eine Funktion $u_i(X)$, die die Zugehörigkeit im Sinne der Fuzzy Logik des Objektes X zur Klasse i beschreibt [Bez73]. In der Erweiterung [Yan93] werden zusätzlich Parameter $0 < \alpha_i < 1$ eingeführt, die für den Anteil der Objekte aus der Klasse i an der Gesamtverteilung stehen. Eine Klassifikation ist dann optimal, wenn der Fehler

$$E_{m,w}(\{X_1, \ldots, X_n\}, V, \alpha) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_i^m(X_j)(d_{i,j}^2 - w \ln(\alpha_i)) \tag{2.1}$$

minimal ist, wobei $d_{i,j}$ der Abstand von V_i und X_j ist und

$$\sum_{j=1}^{n} u_i^m(X_j) = \sum_{i=1}^{c} \alpha_i = 1$$

gelten muß. m bzw. w sind Parameter des Kriteriums, m beschriebt, wie "fuzzy" die Cluster sein sollen und w die Stärke der Bestrafung zu großer Cluster.
Dies entspricht für $m = 1$ und $w = 0$ genau dem Varianzkriterium

$$E(\{X_1, \ldots, X_n\}, V, \alpha) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_i(X_j)(d_{i,j}^2) \tag{2.2}$$

woraus nach [SL77] folgt, daß

$$\forall i : \exists j : \forall k : u_j^m(X_i) = 1 \wedge j \neq k \rightarrow u_k^m(X_i) = 0 \tag{2.3}$$

Für $m > 1$ und $w = 0$ wird diese Formel auch als das Fuzzy Funktional bezeichnet und obige Eigenschaft für die $u_i(X_j)$ gilt im Allgemeinen nicht.

Bemerkung 2.1 (Deutung als Maximum Likelihood Schätzer, [Yan93])
Unter den Voraussetzungen

- *daß alle Cluster gleichwahrscheinlich sind*
- *die Cluster als Normalverteilung mit gleicher Varianz und Covarianz die zu clusternden Beispiele erzeugt haben*

stellt ein Schätzer, der $E(\{X_1, \ldots, X_n\}, V, \alpha)$ minimiert, einen Maximum Likelihood Schätzer für diese Verteilung dar.

Der Teilterm $w \ln(\alpha_i)$ bei $w > 0$ in Gleichung 2.1 stellt einen Versuch dar, die Annahme der Gleichwahrscheinlichkeit der Cluster aufzuweichen [Yan93] und bestraft zu große Cluster (daher wird dieser Term auch als *Penalty* bezeichnet).

2.1.2 Direktes diskretes Clusterverfahren

Für die Parameterwahl $m = 1$ und $w = 0$ reduziert sich das Kriterium 2.1 auf das Varianzkriterium 2.2. Das als K-means Clustering bekannte Verfahren versucht, ausgehend von einer Anfangsaufteilung der Daten, die Klassifikation nach jedem Schritt zu verbessern, ohne allerdings ein Kriterium zu beachten. In [SL77] wird dagegen ein *Hill-climbing*-Verfahren beschrieben, das sich direkt am Varianzkriterium orientiert und dieses in jedem Schritt, der die Gruppierung verändert, echt

verbessert. Da die Fehlerfunktion nach unten hin beschränkt ist, wird eine Zyklenbildung im Verfahren vermieden. Da durch diese Form der Suche nach besseren Lösungen lokale Minima der Fehlerfunktion erreicht werden können, wird diesen Verfahren meist eine Phase nachgeschaltet, die mithilfe globalerer Verfahren versucht, das Varianzkriterium zu verbessern. Insgesamt ergibt sich ein dreistufiges Verfahren, daß durch das Program *ccluster*, daß im Rahmen dieser Arbeit erstellt wurde, implementiert wird:

1. Bestimme eine Anfangspartition.
2. Verbessere diese Partitionierung durch ein *hill-climbing* Verfahren
3. Verbessere die Partitionen mit einem *split-join-Verfahren* (siehe Algorithmus 2.4), gehe wieder zu 2 oder breche ab.

Verbesserung der Anfangspartition mit hill-climbing

Algorithmus 2.2 (Direktes diskretes Clusterverfahren, [SL77])

1. *Wähle eine Anfangspartition*
2. *Breche ab, wenn die Klassifikation hinreichend genau ist*
3. *Für alle Elemente X_i in beliebiger Reihenfolge:*
   Sei X_i dem Cluster j zugeordnet ($u_j(X_i) = 1$) und für $j \neq k$ gelte

$$q \cdot \frac{count_k}{count_k + 1} \|X_i - V_k\|^2 < \frac{count_j}{count_j - 1} \|X_i - V_j\|^2 \tag{2.4}$$

   Dann ordne das Element X_i dem Cluster k zu (Elementaustausch)[1].
4. *Gehe nach 2*

mit $count_i = \sum_{j=1}^{n} u_i(X_j)$ *und* $V_i = \frac{\sum_{j=1}^{n} u_i(X_j) X_j}{\sum_{j=1}^{n} u_i(X_j)}$ *und* $q \geq 1$ *ist frei wählbar.*

In der vorliegenden Implementierung wurde das Verfahren abgebrochen, wenn kein Elementaustausch mehr möglich ist. Falls für ein Element mehr als ein Elementaustausch möglich ist, wird der Elementaustausch gewählt, bei dem sich das Kriterium am stärksten reduziert. Eine erhebliche Reduzierung des Berechnungsaufwandes ist bei hochdimensionalen Vektoren möglich, da man vor der Berechnung des Abstandes von zwei Vektoren stets bestimmen kann, wie groß der Vektor höchstens sein darf, damit das Ergebnis dieser Abstandsberechnung einen Einfluß auf die folgenden Schritte im Verfahren hat. Man kann daher während der Abstandberechnung regelmäßig prüfen, ob diese Schranke schon überschritten wurde und abbrechen, wenn dies der Fall ist (gedeckelte Abstandsberechnung). [2]

Zur Beschleunigung und der numerischen Stabilisierung des Verfahrens wird bei allen Versuchen $q = 1.05$ gewählt, sofern nichts anderes erwähnt wird. Diese Parameterwahl kann man so interpretieren, daß die Verbesserung der Varianz bezüglich der zwei betrachteten Cluster mindestens 5% betragen muß. Für $q = 1$ erhält man das klassische Verfahren.

[1] Ein Elementaustausch unter dieser Voraussetzung führt zu einer echten Verbesserung des Kriteriums 2.2.

[2] Die Optimierung "gedeckelte Abstandsberechnung" ist auf herkömmlichen Einprozessor- und MIMD Maschinen sehr effizient und ist in der Regel nicht auf SIMD Maschinen übertragbar. Die Implementierung der selbstorganisierenden Karten, die auf den SIMD Rechnern *MASPAR* und *CNAPS* begonnen wurde, konnte nach der Einführung dieser Optimierung nicht mehr deutlich bessere Laufzeiten als die entsprechende Implementierung auf einer Workstation erzielen. Dies ist besonders deshalb verwunderlich, da bei der *MASPAR* 50% der theoretisch möglichen Fließkommaleistung der Maschine ausgeschöpft wurden, bei der *CNAPS* immerhin 10% der denkbaren Spitzenleistung.

Finden der Ausgangspartition Für das Auffinden der Anfangspartition kann man verschiedene Verfahren wählen. Die Wahl einer guten Ausgangspartition kann sowohl eine erhebliche Beschleunigung des Verfahrens als auch eine deutlich Verbesserung des Ergebnisses bewirken [SL77], die unten beschriebenen Verfahren sind in *ccluster* implementiert.

Zufallszuweisung Wähle für jedes Element zufällig einen Cluster aus [SL77].

Buckshot Wähle aus den n vorhandenen Beispielen zufällig $\sqrt{c \cdot n}$ aus und clustere diese auf c Cluster. Wähle die Clustermittelpunkte gemäß dieser Cluster und ordne die Beispiele jeweils dem nächsten Clustermittelpunkt zu [SL77].

Fractioning Wähle eine Reduktionsfaktor p und eine Samplegröße k. Die Trainingsdaten werden in Mengen der Größe k aufgeteilt und mit einem Clusterverfahren werden jeweils k/p Klassen pro Menge bestimmt. In der nächsten Iteration werden nur die Vektoren beibehalten, die in ihrer Klasse dem Klassenmittelpunkt am nächsten liegen.

Initialisierung mit bekannten Werten Wenn bereits aus anderen Gründen eine "natürliche" Anfangspartitionierung vorgegeben ist, kann man diese verwenden. Bei dem später vorgestellten Verfahren erfolgt eine iterative Schätzung von Wortklassen und die letzte Schätzung liefert eine gute Anfangspartition (siehe Bemerkung 3.9).

Falls eine *Initialisierung mit bekannten Werten* möglich ist, sollte diese auf jeden Fall verwendet werden. Die Verfahren *Zufallszuweisung*, *Buckshot* und *Fractioning* werden in dieser Reihenfolge immer besser und zeitaufwendiger[3].

Verbesserung der Lösungen mit einem split-join Verfahren Das Verfahren zur Verbesserung der Ausgangspartition hat die Eigenschaft, sich leicht in lokale Minima zu bewegen. Eine Asymmetrie des Verfahrens liegt darin begründet, daß ein Element, das als einziges einem Cluster zugeordnet ist, nicht in einen anderen Cluster wandern kann. Dieser Extremfall trat tatsächlich sehr oft bei der Ursprungsversion von *ccluster* auf und führte zu entsprechend unnatürlichen Resultaten.

Dieses und andere Probleme werden von einem in *ccluster* implementierten Verfahren beseitigt, daß zunächst zwei Cluster vereinigt (join) und anschließend ein Cluster in zwei Teile aufteilt (split), sofern sich damit die Fehlerfunktion echt verbessert. Damit bleiben die günstigen Konvergenzeigenschaften des obigen inkrementellen Verfahrens erhalten und die Verfahren können beliebig gemischt werden. Das komplizierte zweistufige Verfahren ist notwendig, damit man die Anzahl der Klassen nicht verändert und der Wert der Fehlerfunktion vergleichbar bleibt.

Mit dem nachfolgenden Kriterium kann man die Veränderung der Varianz durch eine join-Operation direkt aus den Clustermittelpunkten bestimmen, ohne die einzelnen Elemente betrachten zu müssen. Die Varianzänderung durch eine split-Operation kann man bestimmen, indem man die Join Varianz der Zusammenfassung der aufgespaltenen Cluster zu einem Cluster berechnet. Dies führt dann durch das folgende Theorem direkt auf ein Verfahren.

Theorem 2.3 (Join Varianz auf Clustermittelpunkten)
Die Varianz enstprechend des Varianzkriteriums wird bei der Vereinigung der Klas-

[3] Bei der Implementierung *ccluster* wird die sinnvolle Anwendbarkeit der verschiedenen Initialisierungsverfahren bezüglich der Größe des Clusterproblems genau kontrolliert, da aufgrund des rekursiven Charakters der Initialisierung viele Clusterprobleme gelöst werden müssen, für die jeweils eine Anfangspartition gebildet werden muß. Für sehr kleine Clusterprobleme ist z.B. nur das Verfahren der Zufallszuweisung sinnvoll.

sen i und j um

$$\frac{\|V_i - V_j\|^2}{\frac{1}{\text{count}_i} + \frac{1}{\text{count}_j}}$$

erhöht, wobei V_i, V_j die jeweiligen Klassenmittelpunkt und $\text{count}_i, \text{count}_j$ *die Anzahl der Elemente in den jeweiligen Klassen darstellt.*

Beweis:
OBdA kann man annehmen, daß das Problem vor dem join nur 2 Klassen enthielt und $i \neq j$. Sei W die *within-group-scatter-matrix*, B die *between-group-matrix* und T die *total-scatter-matrix*, dann gilt $T = W + B$ [SL77, Seite 34]. W und B sind von der Klasseneinteilung abhängig, T ist es nicht. Das Varianzkriterium läßt sich damit auch als $spur(W) = spur(T) - spur(B)$ schreiben [SL77, Seite 102]. Da beim Einklassenfall $spur(B') = 0$ und folglich $spur(W') = spur(T)$ gilt ist der Zuwachs durch den join durch $spur(B)$ gegeben und Ausrechnen für den Spezialfall führt auf oben angegeben Term. □

Algorithmus 2.4 (Split-join Verfahren)
Um einen Split-Join Schritt durchzuführen sind folgende Schritte nötig

1. *Wende ein Clusterverfahren auf die Clustermittelpunkte an.*
2. *Berechne innerhalb der Cluster von Clustermittelpunkten die Join Varianz zwischen allen Paaren von Clustermittelpunkten und ordne die Paare von Clustermittelpunkten nach der Größe der Join Varianz.*
3. *Berechne die Split Varianz jedes Clustermittelpunktes.*
4. *Falls es eine Join Varianz gibt, die echt kleiner ist als eine Split Varianz, dann führe den entsprechenden Split-Join durch.*

Für eine Implementierung wird man bei der iterierten Anwendungen des Split-Joins versuchen, möglichst viele der hier berechneten Werte inkrementell abzuändern, was bis zu einem gewissen Grad hin möglich ist. Das Clustern der Clustermittelpunkte ist nötig, da sonst quadratisch viele Abstandsberechnungen zwischen den Clustermittelpunkten nötig wären. Man beachte, daß die Join Varianz bis auf den Nenner der euklidischen Distanz entspricht und man erwarten kann, daß sich in einem Cluster von Clustermittelpunkten solche mit geringer Join-Varianz befinden.

2.1.3 Penalized Fuzzy-C-Means

Der Penalized Fuzzy-C-Means (PFCM) Algorithmus nach [Yan93] orientiert sich ebenfalls an der Fehlerfunktion 2.1. Er stellt im Gegensatz zum vorherigen Algorithmus allerdings keine besonderen Anforderungen an m und w. Durch diese Abschwächung der Vorbedingung ist es nicht mehr möglich, die Parameter nach jedem Trainingsbeispiel zu verändern, sondern erst, nachdem alle Beispiele einmal gesehen wurden. Die wichtige technische Optimierung durch die "gedeckelte Abstandsberechnung" ist hier ebenfalls nicht möglich, sodaß eine Implementierung auf *SIMD* Rechnern noch Vorteile verspricht (siehe auch [Gün91] für eine Argumentation für die Implementierung auf Vektorrechner). Diese zu [Yan93] etwas geänderte Darstellung soll in erster Linie dokumentieren, daß eine effiziente Berechnung dieses Verfahrens möglich ist, die außer der oben gemachten Einschränkung kaum weniger effizient ist als ein herkömmliches direktes Verfahren. In den implementierungsorientierten Darstellungen [CDB86][Gün91] werden andere, wesentlich ineffizientere Berechnungsfolgen vorgeschlagen als die hier entwickelte. Ein auf der hier beschriebenen Berechnungsfolge basierendes Verfahren ist in *ccluster* implementiert und

kann nach einer Anfangsklassifikation mit einem diskreten Verfahren aus dem Abschnitt 2.1.2 gestartet werden[4].

Zu Beginn einer Iteration seien für alle Cluster i die Analoga zum Klassenmittelpunkt V_i und zur Anzahl der Elemente in der Klasse $count_i$ bekannt

$$count_i = \sum_{j=1}^{n} u_i^m(X_j) \qquad V_i = \frac{\sum_{j=1}^{n} u_i^m(X_j) \cdot X_j}{count_i}$$

Der Algorithmus berechnet iterativ folgende Gleichungen für alle X_j:

$$\alpha_i = \frac{count_i}{\sum_{j=1}^{c} count_j} \tag{2.5}$$

$$u_i^*(X_j) = \sqrt[m-1]{\frac{\|X_j - V_j\|^2 - wln(\alpha_i))}{q}} \tag{2.6}$$

$$u_i(X_j) = \frac{u_i^*(X_j)}{\sum_{k=1}^{c} u_i^*(X_j)} \tag{2.7}$$

Damit kann man die Werte

$$count_i = \sum_{j=1}^{n} u_i^m(X_j) \qquad V_i \cdot count_i = \sum_{j=1}^{n} u_i^m(X_j) \cdot X_j$$

inkrementell berechnen, ohne sich die anderen Ergebnisse von Rechnungen für andere $X_{j'}$ merken zu müssen. Bei der Berechnung von $u_i^*(X_j)$ kann $q > 0$ beliebig gewählt werden, ohne $u_i(X_j)$ in seiner Größe zu ändern. Aus Gründen der numerischen Stabilität ist die Wahl $q = max_i|\|X_j - V_j\|^2 - wln(\alpha_i)|$ sinnvoll.

2.2 Lineare und nichtlineare Projektionen

Methoden zur linearen und nichtlinearen Projektion werden für unterschiedliche Aufgaben benötigt. Dazu gehören insbesondere die Vorverarbeitung für einen komplexeren Klassifikator und die Visualisierung in einem für Menschen interpretierbaren Vektorraum (z.B. $\mathcal{R}^2$). Diese Methoden sind also insbesondere als Vorverarbeitung für neuronale Parser und zur Visualierung geeignet.

2.2.1 Lineare Methoden zur Informationsreduktion

Unter linearen Methoden sollen solche Methoden verstanden werden, die eine Informationsreduktion durch eine orthonormale Projektion der Matrix in einen niedrigdimensionalen Raum herstellen.

Singulärwertzerlegung Die Singulärwertzerlegung ist eine klassische mathematische Methode zur Untersuchung von großen Datenmengen und hat daher einen hohen Stellenwert bei der Untersuchung von Corpora. Wichtige Untersuchung in diesem Bereich sind [Sch92] [Sch93b][Sch93c][Sch93d]. Weitere wichtige Anwendungsgebiete finden sich im Information Retrival[5] und in der Signalverarbeitung. Die dabei verwendeten Techniken zur Singulärwertzerlegung großer, dünnbesiedelter Matrizen sind in [BDO+93] beschrieben und in dem dazugehörigen frei verfügbaren Softwarepaket *SVDPACKC* implementiert. Die effiziente neuronale Berechnung stark verwandter Größen wird in [Dia92] beschrieben, [HH89] stellt allgemeine mathematische Grundlagen dar.

[4] Das Skript *fuzzy* kann zur Visualisierung der Fuzzy-Klassifikation benutzt werden.

[5] Information Retrival ist ein Wissenschaftsgebiet, daß sich mit der Selektion von Textpassagen nach inhaltlichen Kriterien beschäftigt.

Definition 2.5 (Singulärwertzerlegung) *Sei A eine $m \times n$ Matrix mit $rang(A) = r$ und $A = U\Sigma V^T$ mit $UU^T = VV^T = I_n$, $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ und $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r = \sigma_{r+1} = \ldots = \sigma_n = 0$. $\langle u_i, \sigma_i, v_i \rangle$ ist das i-te Singulärwerttripel, wobei u_i bzw. v_i die i-te Spalte der Matrix U bzw. V sind. Die σ_i sind die Singulärwerte.*

Daraus folgt nach [BDO+93] und [HH89]

Beobachtung 2.6

- *u_i bzw. v_i sind Eigenvektoren der Matrizen AA^T bzw. A^TA zum Eigenwert σ_i^2*
- *jede Matrix A besitzt eine Singulärwertzerlegung*
- *die Matrix A besitzt die dyadische Zerlegung $A = \Sigma_{i=1}^{n} u_i \sigma_i v_i^T$*

Eine der Kerneigenschaften der Singulärwertzerlegung besteht darin, daß man in der dyadischen Zerlegung kleine Singulärwerte σ_i weglassen kann, ohne die "wesentlichen" Informationen über die Matrix A zu verlieren.

Definition 2.7 (Normen) *Die Matrixnormen $\|\cdot\|_2$ bzw. $\|\cdot\|_F$, die Zeilenbetragssummennorm bzw. die Fröbeniusnorm, sind definiert als*

- $\|A\|_2 = \sqrt{\overline{A}^T A}$
- $\|A\|_F = \sqrt{spur(\overline{A}^T A)}$

Die Vektornorm $\|x\|_2$ ist die übliche euklidische Distanz.

Nach [BDO+93] und [HH89] gilt

Theorem 2.8 (Informationsreduktion) *Sei A eine $m \times n$ Matrix mit $rang(A) = r$ und $\langle u_i, \sigma_i, v_i \rangle$ das i-te Singulärwerttripel. Sei außerdem $A_k = \Sigma_{i=1}^{k} u_i \sigma_i v_i^T$ mit $k < r$, dann gilt*

- $\min_{\text{rang}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$
- $\min_{\text{rang}(B)=k} \|A - B\|_F^2 = \|A - A_k\|_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_r^2$
- $\|Ax\|_2^2 \leq \|A\|_F^2 \|x\|_2^2$

Aus der Definition von A_k kann direkt eine orthonormale Projektion in einen Unterraum abgeleitet werden.

Aus dem obigen Theorem kann man folgern, daß man anhand der Größe eines Singulärwertes abschätzen kann, wie stark er auf die Wiedergabe der Matrix zurückwirkt. Es stellt zudem klar, daß diese Form der Reduktion eine bestmögliche bezüglich von Matrizen des entsprechenden Ranges ist. Es stellt sich also die Frage, wie groß k sein sollte, um einen guten Kompromiß zwischen Informationsreduktion und Wiedergabetreue zu erhalten. Dazu gibt es die Kriterien

- das Verhältnis $\frac{\sigma_k}{\sigma_{k+1}}$ unterschreitet eine gewählte Schranke
- σ_k ist genügend klein (sehr viel kleiner als 1)

Zufallsprojektion Die Singulärwertzerlegung liefert eine mathematisch gut begründete Auswahl einer orthonormalen Projektion. Falls aber der Ausgangsraum extrem groß ist, kann diese in der Praxis möglicherweise nicht berechnet werden. In [RK89] wird daher eine Zufallsprojektion durchgeführt und gezeigt, daß sich der dadurch verursachte Fehler in gewissen Grenzen hält, sofern der Unterraum eine genügend große Dimension besitzt:

Theorem 2.9 *Sei Φ eine $d \times D$ Matrix, deren Einträge mit einer isotropen Verteilung gewählt wurden und deren Spalten normiert sind und $\langle\cdot\rangle_\Phi$ der Erwartungswert einer Zufallsgöße unter dieser Wahl von Φ. Dann gilt für alle Vektoren $x, y \in \mathcal{R}^D$ die Relation*

$$\langle(\|\Phi x - \Phi y\|_2^2 - \|x - y\|_2^2)^2\rangle_\Phi \leq \frac{2}{d} \cdot \|x - y\|_2^4$$

2.2.2 Neuronale und nichtlineare Methoden zur Informationsreduktion

Die verwendeten Analysemethoden zur Analyse einzelner Repräsentationen gehören zur Klasse der unüberwachten Lernverfahren. Der genaue technische Hintergrund der neuronalen Verfahren kann z.B. in [Roj93] nachgelesen werden.

Multidimensionale Skalierung Ein wichtiges Verfahren der Multidimensionale Skalierung ist die nichtlineare Abbildung nach [Sam69], auch oft nach dem Autor als *Sammon's Mapping* benannt. Ziel der *Sammon's Mapping* ist es, eine Menge von Vektoren in einem hochdimensionalen Raum so auf eine Menge von Vektoren abzubilden, daß die Abstände zwischen den Vektoren erhalten bleiben. Seien X_i der Ausgangsvektor und Y_i der entsprechende Zielvektor sowie $d^*(\cdot,\cdot)$ und $d(\cdot,\cdot)$ Abstandsmaße auf dem Ausgangsraum bzw. Zielraum, dann ist eine Abbildung gesucht, die folgenden Fehler minimiert

$$E = \frac{1}{\sum_{i<j} d^*(X_i, X_j)} \sum_{i<j} \frac{(d^*(X_i, X_j) - d(Y_i, Y_j))^2}{d^*(X_i, X_j)}$$

Aus dieser Fehlerfunktion läßt sich insbesondere für die euklidische Distanz eine Gradientenabstiegsmethode ableiten, die solche Abbildungen lernt. Diese Methode eignet sich gut, um sich ohne Vorwissen einen ersten Überblick über ein Problem zu verschaffen. Sie kann insbesondere helfen, die Größe einer selbstorganisierenden Karte nach Kohonen zu bestimmen, deren Neuronen im Zielraum R^n liegen. Eine frei verfügbare Implementierung ist im Rahmen des Paketes *SOM-PAK* gegeben, daß an der Universität Helsinki entwickelt wurde.

Autoassoziatoren Ein Autoassoziator ist ein neuronales Netz, das lernen soll, die Eingabe an der Ausgabe zu reproduzieren. Diese Aufgabe wird dadurch erschwert, daß das Netz die Informationen nicht direkt von der Ein- zur Ausgabe übertragen kann. In dieser Arbeit werden ausschließlich dreischichtige Netze verwendet, wie in Abbildung 2.1 gezeigt. Dieses Netz enthält in der inneren Schicht eine geringere Anzahl an Neuronen als in der Ein- bzw. Ausgabeschicht. Die Aktivierung der Neuronen der inneren Schicht kann von daher als eine Repräsentation der Eingabe in einem niedrigdimensionalen Raum betrachtet werden. Formal betrachtet soll in dieser Arbeit ein Autoassoziator immer wie folgt verstanden werden

Definition 2.10 (Autoassoziatior) *Seien $U, V \in \mathcal{R}^{n \times r}$ und $act : \mathcal{R} \mapsto \mathcal{R}$ und $act' : \mathcal{R}^r \mapsto \mathcal{R}^r$ die komponentenweise Erweiterung von act. Ein Autoassoziator $\langle n, r, U, V, act\rangle$ ist eine Funktion, die einem Vektor $v \in \mathcal{R}^n$ einen Vector $v'' =$*

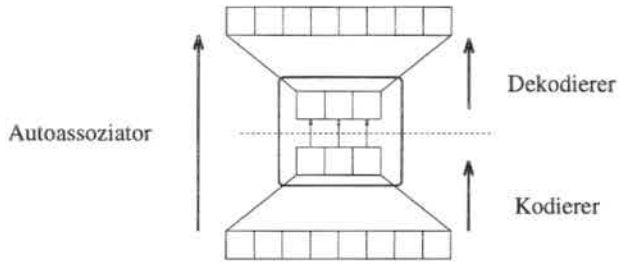


Abbildung 2.1: **Autoassoziator zur Informationsreduktion:** Ein Autoassoziator läßt sich in einen Kodier- und eine Dekodierteil aufspalten, indem man die Ausgaben der inneren Schicht (Kodierung) in ein einschichtiges Netz einspeist, das gerade aus dem Teil des Netzes nach der inneren Schicht besteht (Dekodierung).

$Uv' \in \mathcal{R}^n$ mit $v' = act(V^T v) \in \mathcal{R}^r$ derart zuordnet, daß für eine Trainigsmenge $v_1, \ldots, v_m$ unter der angegebene Wahl von U und V die Summe $\sum_{i=1}^{m}(v_i - v_i'')^2$ bzw. eine andere Fehlerfunktion, die die Güte der Approximation von v_i durch v_i'' geeignet beschreibt, minimal (bzw. genügend klein) wird. Der Vektor v' wird auch Kodierung von v genannt. Zur Informationsreduktion ist $r < n$ erforderlich.

Wenn *act* die Identität ist, wird eine der Singulärwertzerlegung vergleichbare Informationsreduktion erzeugt, wie man bereits anhand der Ähnlichkeit der Formulierungen erkennen kann. Wird *act* dagegen als Sigmoidfunktion gewählt, erhält man eine nichtlineare Methode zur Informationsreduktion, wie sie auch in dieser Arbeit genutzt wird. Als Beispiel für die Mächtigkeit von Autoassoziatoren wird häufig die Entwicklung eines Binärkodes bei geeigneter Eingabe angeführt. Auf die weiteren Standardanwendungen für Autoassoziatoren sei auf die einschlägige Literatur hingewiesen. In [Pol90][Ber91] wird eine für die Sprachverarbeitung interessante Anwendung von Autoassoziatoren, den RAAM, beschrieben, die im Abschnitt 2.3.2 weiter besprochen werden soll.

Selbstorganisierende Karten Die selbstorganisierenden Karten (SOM)[6] sind eine effiziente und lang untersuchte Methode, die in [RSM90] ausführlich dargestellt wird. Die in diesem Abschnitt zitierten theoretischen Resultate und Methoden lassen sich dort alle vertieft nachlesen, die auf Anschaulichkeit orientierte Darstellung hier orientiert sich an [Roj93]. Eine frei verfügbare Implementierung ist im Rahmen des Paketes *SOM-PAK* gegeben, daß an der Universität Helsinki entwickelt wurde.

Die Grundidee bei der Erstellung einer SOM ist die "Kartierung" eines Ausgangsraumes A in einem Zielraum B mit einer Funktion f, d.h. die Suche nach einer Funktion f mit Definitionsbereich A und Wertebereich B. Sei darüberhinaus p eine Wahrscheinlichkeitsdichte über A und für jeden Punkt § des Ausgangsraumes sei $p(\S)$ endlich. Natürlich gibt es sehr viele solcher Funktionen f, so daß man zusätzlichen Forderungen an eine solche Kartierung stellt

- die Kartierung soll topologieerhaltend sein, d.h. zwei in A nahe beieinanderliegende Punkte sollen auch in B nahe beieinanderliegen
- Gebiete mit vielen (bzw. wenigen) Beispielen in A sollen durch große (bzw. kleine) Gebiete in B kartiert werden,

[6] In der englischen Literatur wird oft der Begriff *Kohonen feature maps*, *Self-Organizing Semantic Maps* (SOM) gebraucht.

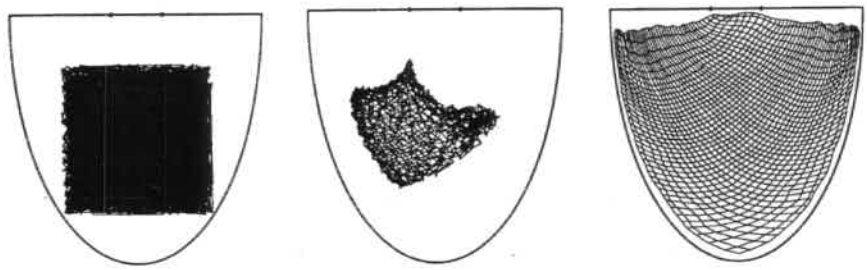


Abbildung 2.2: **Kartierung eines Merkmalsraumes mit SOM:** Eine SOM bildet eine selbstorganisierende Karte eines gleichverteilten Ausgangsraumes (halbes Oval) in einem Zielraum (quadratisches Gitter) aus. Die Abbildungen nach [RSM90, Seite 63] zeigen die Kanten zwischen den Gewichten im Zielraum benachbarter Neuronen während des Lernverfahrens.

- f soll durch einen möglichst einfachen und effizienten Algorithmus bestimmt werden
- f soll, wenn es bestimmt ist, effizient berechnet werden können

Tatsächlich ist die theoretische Durchdringung dieser Eigenschaften für den nun folgenden Algorithmus nur in engen Grenzen gegeben – einige wenige wichtige Resultate konnten durch die Theorie der Markov Prozesse gewonnen werden. Seine Rechtfertigung erfolgte daher vielmehr durch eine intensive empirische Forschung, die häufig durch geschickte Visualisierungen wie in Abbildung 2.2 zusätzliche Plausibilität erhalten. Für viele technische Probleme sind in [RSM90] effiziente Lösungen, die auf Erweiterungen dieses Algorithmus basieren, angegeben. Ebenfalls in [RSM90] sind eine Reihe von neurobiologischen Phänomenen durch SOM erklärt worden. Zusätzliche Plausibilität erfährt der Algorithmus durch die Ähnlichkeit mit bekannten Clusterverfahren.

Algorithmus 2.11 (Lernverfahren für Selbstorganisierende Karten)
Sei $\epsilon_t \overset{t\to\infty}{\longrightarrow} 0$ die Schrittweite, $w_1, \ldots, w_m \in \mathcal{R}^n$ die Gewichte der Neuronen und $h_t(i,k) \overset{t\to\infty}{\longrightarrow} 0$ ein Abstandsmaß auf den Neuronen, die im Zielraum B liegen. Der Ausgangsraum A ist der $\mathcal{R}^n$.

1. *Wähle $w_1^0, \ldots, w_m^0 \in \mathcal{R}^n$ zufälllig* [7] *und $t = 0$*
2. *Wähle entsprechend p zufällig ein Beispiel $\S^t$ aus A*
3. *Ermittele das Neuron i, sodaß der Abstand $-V_t(i)$ von $\S^t$ minimal wird*
4. *Setze $w_i^{t+1} := w_i^t + \epsilon(t) \cdot h_t(i,k) \cdot (\S^t - w_i^t)$*
5. *Inkrementiere t und gehe nach 2*

In diesem Verfahren sind noch einige Parameter unbestimmt, insbesondere ist der Abbruch des Verfahrens unzureichend beschrieben, $V_t(i)$, ϵ_t und $h(i,k)_t$ sind weitgehend frei wählbar.

In B stellt man sich häufig zunächst eine Zuordnung der Neuronen auf ein zweidimensionales, quadratisches Gitter vor und indiziert die Neuronen mit den

[7] In manchen Anwendungen gibt es eine sinnvolle Anfangsbelegung, was aber im Allgemeinen nicht der Fall zu sein braucht.

Gitterpunkten, sodaß man mit $\sigma_t \stackrel{t\to\infty}{\longrightarrow} 0$ und $h_t(i,k) = e^{-\frac{1}{2}\cdot(\frac{i-k}{\sigma_t})^2}$ die geforderten Eigenschaften erhält. Gewöhnlich wird darüberhinaus $V_t(i) = -w_i^t \cdot \S^t$ oder $V_t(i) = -\frac{1}{2}\cdot\sum_{j=1}^{n}(w_{ij}^t - \S_j^t)^2$ gewählt. Der Abbruch des Verfahrens wird bei einem sehr kleinen σ_t vorgenommen. Stark vereinfacht gilt:

Theorem 2.12 (Konvergenz der SOM, [RSM90]) *Sei ϵ_t hinreichend klein für alle t, alle w_i bei der Initialisierung nahe genug an einem Gleichgewichtszustand, dann sind die folgenden Bedingungen notwendig und hinreichend für die "Konvergenz" des Verfahrens*

$$\lim_{t\to\infty}\int_0^t \epsilon_{t'}dt' = \infty$$

$$\lim_{t\to\infty}\epsilon_t = 0$$

Nach diesem Theorem wären also insbesondere alle Funktionen der Form $\epsilon_t \propto t^{-\alpha}$ mit $0 < t \leq 1$ möglich. In vielen praktischen Fällen kann aber die erste Bedingung abgeschwächt werden zu $\lim_{t\to\infty}\int_0^t \epsilon_{t'}dt' \gg 1$ [RSM90].

Prinzip der Vorhersagbarkeitsminimierung Während die bisherigen Kodierungen nur indirekte Maßstäbe zur Erreichung des Zieles einer möglichst informativen Darstellung der Daten darstellen, wird in [Sch93a] das *Prinzip der Vorhersagbarkeitsminimierung* eingeführt. Eine Repräsentation eines n-dimensionalen Vektors sollte stets so geartet sein, daß jede einzelne Komponente von den verbleibenden statistisch unabhängig ist. Dieses Kriterium kann man auch als Redundanzelimination bezeichen und kann für binäre Codes formalisiert werden als

$$E(y_i|y_k, k \neq i) = E(y_i)$$

Man könnte dies für nicht-binäre Codes auch formalisieren zu

$$H(Y_i|Y_k, k \neq i) = H(Y_i)$$

oder als Fehlerfunktion

$$\sum_i H(Y_i|Y_k, k \neq i)$$

[Sch93a] gibt eine neuronale Implementierung der ersten Gleichung an und erreicht dies, indem er Prediktornetze von je $n-1$ Kodestellen auf die verbleibende Kodestelle trainiert und als Optimimalitätsmaß für die Prediktornetze die optimale Rekonstruktion angibt. Für die Erzeugung des Codes wird ein Netz trainiert, das als Ausgabe den Code hat und als Fehlerfunktion die Abweichung der vom Prediktor vorhergesagten Codes. Auf diese Weise bekommt man einen Wettkampf zwischen zwei Systemen: Die Prediktoren versuchen, die Eingaben genau zu vorherzusagen, das Netz zur Konstruktion versucht diese Vorhersage zu zerstören.

Weitere in ein solches System implementierbare Gütemaßstäbe sind die Rekonstruierbarkeit des Ausgangscodes und die Erhaltung der Wahscheinlichkeitsdichte der Eingabe [Sch93a].

2.3 Neuronale Analyse von strukturierten Folgen

Die Analyse von Folgen von Repräsentationen ist eine Variante der aus der Statistik bekannten Zeitreihenanalyse. Die hier angestrebte Untersuchung soll die Analyse von Wortfolgen ermöglichen und damit die starren Beschränkungen von Ansätzen mit festem Kontext überwinden helfen.

Wie noch im Kapitel 3 deutlich werden wird, kann eine linguistische Einheit bereits durch einen festen Kontext beschrieben werden. Für die Klassifikation von

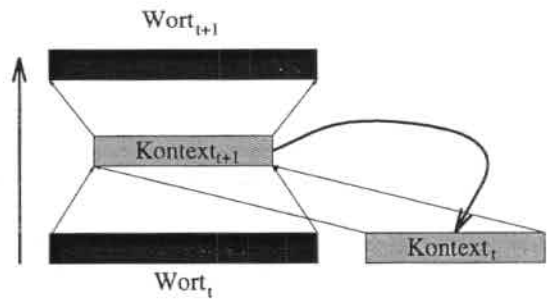


Abbildung 2.3: **Einfach rekursive Netze (SRN) zur Wortprediktion:** Einfach rekursive Netze (SRN) sind dreischichtige Assoziatoren, die dem aktuellen Eingabewort und dem Kontext das nächste Wort zuordnen. Die Aktivierung der inneren Schicht ist der Kontext des nächsten Worts und wird in die Kontextschicht kopiert.

variablen Kontext ist eine Darstellung der Vorgeschichte (bzw. "Nachgeschichte") eines Wortes, eines Satzes, etc. interessant. Solche Darstellungen sind mit den Methoden der Geschichtskompression [Sch93a] mit rekurrenten neuronalen Netzen möglich. Man könnte also hoffen, daß solche Kontexte gute Repräsentationen von linguistischen Einheit darstellen. Insbesondere kann man durch die Analyse des Kontextes nach der Verarbeitung einer Sequenz durch ein Netz ohne Berücksichtigung der Umgebung eine Klassifikation der Sequenz ableiten.

Die Implementierung der hier beschriebenen Netze erfolgte mit dem frei verfügbaren Simulator *xerion*, der an der Universität von Toronto in der Gruppe von Hinton entwickelt wurde. Dazu wurde dieser Simulator u.a. um einen neuen Art von Netzwerkschicht erweitert, einer Ausgabeschicht, deren Ziel es ist, an der Ausgabe genau die Werte einer anderen Schicht zu reproduzieren. Mit einer solchen `ASSOC` Schicht kann man also insbesondere einen Autoassoziator implementieren, der in seinen Trainingsbeispielen nur die Eingabe enthält. Diese Erweiterung erlaubt die einfache Implementierung der RAAM-Architektur.

2.3.1 Einfache rekursive Netze (SRN)

[SSCM91] faßt Elman- [Elm90][Elm91] und Jordan [Jor86] Netze unter dem Oberbegriff des einfach rekursiven Netzes (SRN) zusammen. Ein einfach rekursives Netz ist ein solches Netz, daß eine Menge von Netzwerkschichten besitzt, die sich in *Eingabe*, *Ausgabe*, *Innere* und *Kontext* Schichten aufteilen lassen. Das Netz, das entsteht, wenn man die Kontextschicht wegläßt, ist ein nicht rekurrentes, feed-forward Netz. Eine Kontextschicht ist mit einer inneren Schicht mit festen Gewichten der Stärke 1 Neuron für Neuron verbunden. In aller Regel wird das Netz überwacht mit der Aufgabe trainiert, das nächste Wort vorherzusagen (siehe Abbildung 2.3). [Elm91] zeigt, daß solche SRN in der Lage sind, nichttriviale sprachliche Information in der Kontextschicht abzuspeichern. Die einfach neuronalen Netze haben bezüglich des Trainings besondere Eigenschaften: Man kann eine Teilmenge der Schichten streichen und erhält ein nicht-rekursives Netz, daß von der Eingabe bis zur Ausgabe durchverbunden und daher direkt trainierbar ist. Nach [SSCM91] kann man bei einem SRN zur Wortprediktion zwischen folgenden Lernphasen unterscheiden

1. Das Netzwerk berücksichtigt *kein* Kontextwissen.

2. Die Kontextschicht repräsentiert das vorausgegangene Symbol und dieses wird in den Lernvorgang einbezogen.

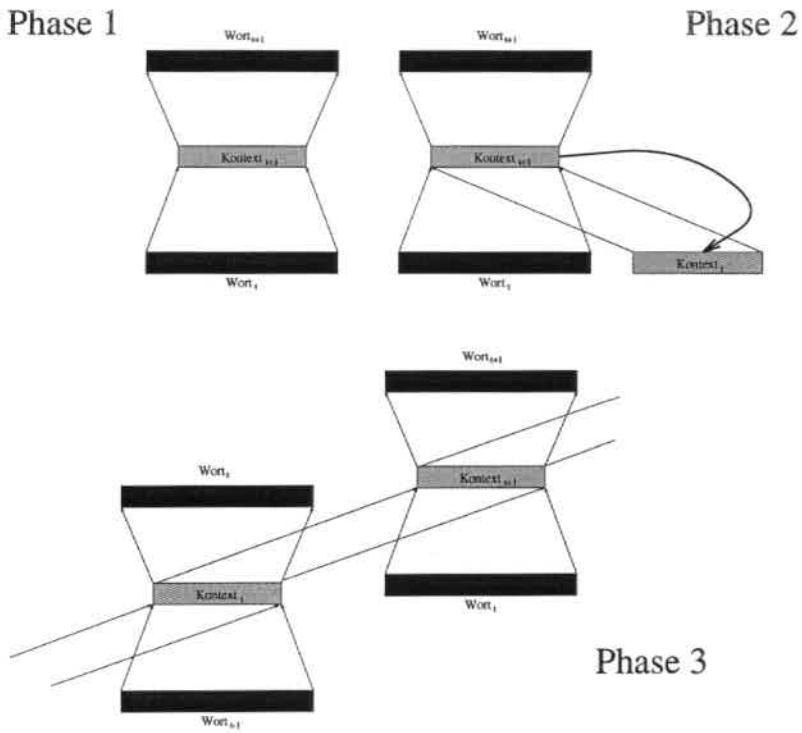


Abbildung 2.4: **Dreiphasenregime für SRN:** In der ersten Phase wird ein Assoziator trainiert, der dem aktuellen Wort das nächste Wort zuordnet. In der zweiten Phase wird dem Assoziator das aktuelle und das letzte Wort präsentiert. In der dritten Phase wird in der Art des Backpropagation Through Time das rekursive Netz abgerollt.

3. Das Netz lernt komplizierte Sachverhalte und die Kontextschicht repräsentiert Wissen, das nicht nur aus dem letzten Symbol besteht.

Daraus resultiert ein Trainingsregime, das sich ebenfalls in drei Phasen gliedert (Abbildung 2.4).

1. Trainiere einen Assoziator, der nur das letzte Wort benutzt.
2. Trainiere das SRN, propagiere den Fehler aber nicht durch die Kontextschicht.
3. Trainiere das SRN mit Backpropagation Through Time.

2.3.2 Der rekursive Autoassoziator (RAAM)

Von [Pol90] wurde der rekursive Autoassoziator RAAM[8] entwickelt, um mit neuronalen Netzen Baumstrukturen verarbeiten zu können (Abbildung 2.5). [Ber91] hat den RAAM benutzt, um aus einer in einem RAAM kodierten Liste von Worten mit neuronalen Netzen einen Analyse im Sinne der Rektions- und Bindungstheorie GB [9] zu extrahieren (Abbildung 2.5). Die Tatsache, daß auf diese Art und Weise eine solch detailierte Analyse möglich war, läßt zunächst einmal den Schluß zu, daß diese Listenrepräsentation die wesentlichen Informationen über die Elemente

[8] Recursive Auto Associative Map
[9] Government and Binding Theory

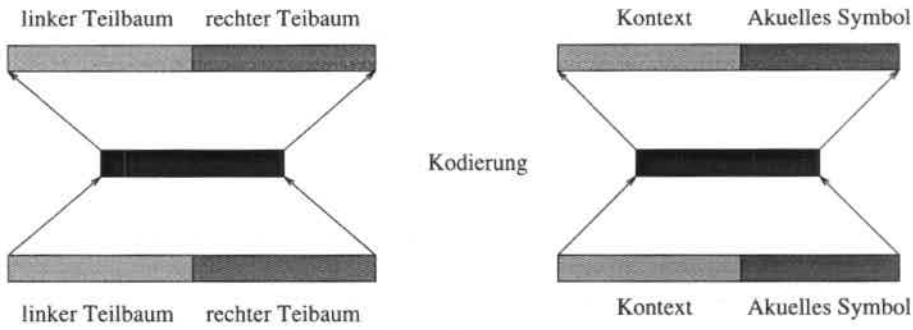


Abbildung 2.5: **Rekursiver Autoassoziator:** Ein Baum kann schrittweise in eine einzige Repräsentation gefaßt werden, indem man rekursiv vorgeht und einen Knoten mit zwei Nachfolgeknoten mit dem oben angegebenen Autoassoziator in einen Knoten verwandelt. Die Dekodierung erfolgt sinngemäß umgekehrt, nur daß das Ende des rekursiven Dekodiervorganges festgestellt werden muß. Während der linke Teil der Abbildung den allgemeinen Fall darstellt, wird der Spezialfall der Kodierung einer Liste im rechten Teil dargestellt. Bei der Kodierung einer Liste muß die Größe der Eingaberepräsentation nicht notwendigerweise mit der der Listenkodierung übereinstimmen und man nennt den linken Teilbaum in Übereinstimmung mit der Sprechweise bei SRN auch Kontext.

der Liste enthält. Andere vielversprechende Experimente sind von [Sch93a] dokumentiert, der die Fähigkeiten von Netzen zum Abspeichern von Eigenschaften über lange Zeiträume untersucht hat. RAAMs haben dabei sowohl eine sehr hohe Speicherkapazität als auch eine ausgesprochen gutes Lernverhalten bewiesen. [Sch93a] trainiert dabei, ebenso wie in dieser Arbeit, stets nur den Autoassoziator und wendet keine Trainingsmethoden für rekurrente Netze an, wie dies in der dritten und aufwendigsten Phase des Dreiphasenregime für SRN der Fall ist.

Gekoppelte RAAM Architekturen und konfluente Analyse In [Chr91] werden zwei RAAM verkoppelt, um ein holistisches Inferenzsystem zu erhalten. Dieses Inferenzystem soll die Aufgabe lösen, eine Abbildung von Sequenzen auf Sequenzen zu lernen. Die Aufgabe soll auf eine holistische Art gelöst werden, d.h. die Abbildung soll in einem Schritt erfolgen und den ganzen Satz auf einmal berücksichtigen. Solche Abbildungen kann man z.B. nutzen, um eine Übersetzung zu erlernen. Die einfachte Möglichkeit, diese Abbildung zu erlernen, besteht darin, jedes RAAM getrennt eine Repräsentation der jeweiligen Sequenz lernen zu lassen und durch ein weiteres neuronales Netz eine Abbildung zwischen diesen Listenrepräsentationen zu etablieren. Solche Abbildungen wurden von [Cha90] zur Passivierung von Sätzen untersucht und diese Architektur soll hier als loose gekoppelt bezeichnet werden (siehe Abbildung 2.6).

Die Idee von [Chr91] bestand nun darin, das Erlernen der Abbildung zu vereinfachen, indem man schon während der des Lernens der Sequenzenrepräsentation die Aufgabe berücksichtigt. Die Grundidee ist dabei, daß am Ende der Präsentation einer Sequenz ein weiterer Backpropagationschritt eingefügt wird. Die erste Variante ist eine Erweiterung der loose gekoppelten Architektur und soll hier als eng gekoppelt bezeichnet werden. Bei dieser Architektur ist das Netz, das die Abbildungen zwischen Listenrepräsentationen lernen soll, schon während des Trainings der RAAM vorhanden. Zum Ende einer Sequenz wird der Fehler, den das Abbildungsnetz erzeugt, in die RAAM der Ausgangssequenz zurückpropagiert. Auf diese

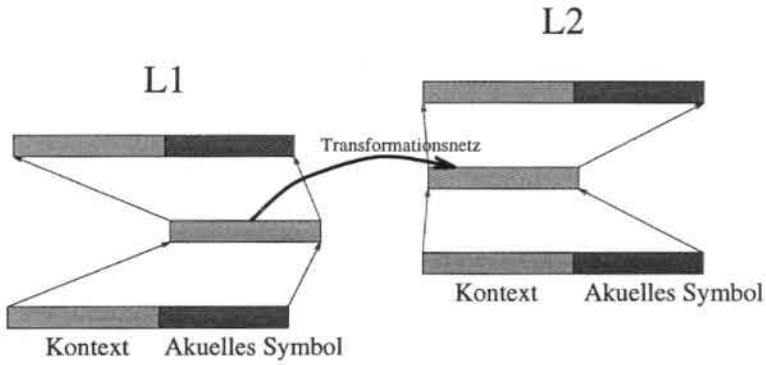


Abbildung 2.6: **Gekoppelte RAAM Architektur** In der gekoppleten RAAM Architektur werden die inneren Schichten zweier RAAM durch ein Verbindungsnetzwerk verbunden. Eine gekoppelte RAAM, deren Verbingsungsnetzwerk während des Lernen der einzelnen RAAM trainiert wurde, wird eng gekoppelt genannt, sonst ist es loose gekoppelt.

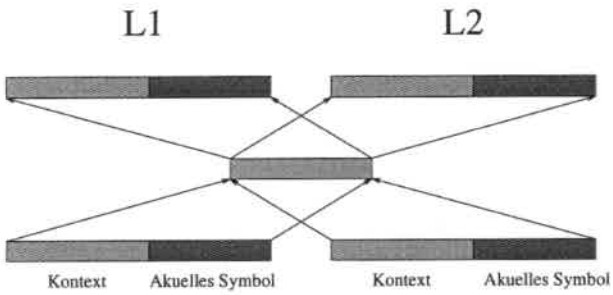


Abbildung 2.7: **Dual-ported RAAM:** Ein dual-ported RAAM ist eine Architektur, bei der in den Sprachen L1 und L2 jeweils die gleiche Repräsentation für die entsprechenden Sequenzen gelernt werden muß.

Weise wird das Ausgangsnetz dazu veranlaßt, Listenrepräsentationen zu erlernen, die sich leichter übertragen lassen. Die andere, in [Chr91] dokumentierte Alternative besteht darin, beide Netze dazu zu zwingen, die gleiche Repräsentation für Sequenzen zu lernen, die einander zugeordnet werden und wird dort als *dual-ported RAAM* bezeichnet. Ein Sequenzenpaar wird dabei auf zwei getrennten RAAMs wie üblich trainiert. Für die Repräsentation, die im letzten Schritt gelernt worden ist, werden jedoch einmal die Decoder der RAAM vertauscht und einmal trainiert. Anschaulich kann man sich auch vorstellen, daß die beiden RAAM Architekturen eine gemeinsame innere Schicht haben (siehe Abbildung 2.7). Die von einer dual-ported RAAM erzeugten internen Repräsentationen können im Rahmen der Übersetzung auch als Interlingua interpretiert werden, für die es von beiden Sprachen her einen Parser und einen Generator gibt. Denkbar wäre hier auch die Erweiterung auf eine n-äre RAAM, die weitere Sprachen hinzunimmt.

Einlernen von Zusatzinformation in ein RAAM-Architektur Im Rahmen der Diplomarbeit werden im Kapitel 3 Methoden vorgestellt, um Sequenzen von Worten zusammenzufassen. Um diese Information in ein Netz einzulernen, kann die RAAM Architektur erweitert werden. Dazu wird eine neue Ausgabeschicht hin-

zugefügt, die mit der inneren Schicht des Netzes verbunden ist und eine Repräsentation der Sequenz assozieren soll, wenn diese vollständig in das Netz eingelesen wurde. Eine besonders wichtige Anwendung des Einlernens von Zusatzinformation ist das Einlernen von Hinweisen, welche Sequenzen von Zeichen komprimiert abgespeichert werden sollen. Dazu präsentiert man am Ende einer als Einheit bekannten Sequenz eine Repräsentation dieser Sequenz.

2.3.3 Selbstorganiserende Karten mit Kontext

Die Temporalen Selbstorganisierenden Karten (TSOM) wurden von [Cha93] eingeführt, um mit Selbstorganisierenden Karten Zeitreihen analysiern zu können. Die wesentliche Idee bei diesem Ansatz ist, die Aktivierung der Neuronen im Algorithmus 2.11 exponentiell zu glätten. Im Schritt 3 wird daher der Abstand V statt bisher $V_t(i) = -\frac{1}{2} \cdot \sum_{j=1}^{n}(w_{ij}^t - \S_j^t)^2$ neu definiert als

$$V_t(i) = d \cdot V_{t-1}(i) - \frac{1}{2} \cdot \sum_{j=1}^{n}(w_{ij}^t - \S_j^t)^2$$

Für $d = 0$ reduziert sich diese Formel auf die alte Abstandsdefinition. Durch schrittweises Erhöhen von d läßt sich der Einfluß des Kontextes erhöhen. In [Cha93] sind sowohl theoretische als auch praktische Belege für die Funktionsfähigkeit diese Ansatzes gegeben worden.

2.4 Entropie und Mutual Information

Dieser Abschnitt soll kurz ein paar in der Arbeit häufig verwendete Definitionen aus der Informationtheorie einführen, die man in [CT91, Kapitel 2] vertiefen kann.

Definition 2.13 *Die* Entropie *einer diskrete Zufallsvariablen X mit dem Wertebereich $\mathcal{X}$ ist definiert als*

$$\mathcal{H}(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2(p(x))$$

Die bedingte Entropie *(auch* Kullback Leibler Distanz *oder* Divergenz *genannt) einer diskrete Zufallsvariablen X mit dem Wertebereich $\mathcal{X}$ ist definiert als*

$$\mathcal{H}(X|Y) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log_2(p(x|y))$$

Die mutual information *zwischen den Zufallsvariablen X und Y ist*

$$\mathcal{MI}(X,Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x) \cdot p(y)}$$

Die relative Entropie *der Wahrscheinlichkeitsdichte p bezüglich q ist*

$$d(p||q) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}$$

Es gelten die folgende Rechenregeln

$$\mathcal{MI}(X,Y) = \mathcal{MI}(Y,X) = \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X,Y) = d(p(x,y)||p(x)q(y))$$

$$\mathcal{H}(X,Y) = \mathcal{H}(X|Y) + \mathcal{H}(Y)$$

Die *Mutual Information* hat die Eigenschaft, daß $\mathcal{MI}(X,Y) \geq 0$ und $\mathcal{MI}(X,Y) = 0$ gdw. X und Y unabhängig sind. Da die *Mutual Information* die Dreiecksungleichung nicht erfüllt handelt es sich bei ihr nicht um eine Metrik. Es gilt, daß $\mathcal{H}(X,Y) - \mathcal{MI}(X,Y) = \mathcal{H}(X|Y) + \mathcal{H}(Y|X)$ eine Metrik ist. Für die *relative Entropie* gilt $d(p||q) \geq 0$ und $d(p||q) = 0$ gdw. $p(x) = q(x)$ für alle $x \in \mathcal{X}$. Die *relative Entropie* ist nicht symmetrisch, $d(p||q) + d(q||p)$ ist zwar symmetrisch, erfüllt aber die Dreiecksungleichung auch nicht.

2.5 Der EM-Algorithmus

Der EM-Algorithmus findet maximum-likelihood Parameterschätzungen auch für solche Modelle, bei denen nicht alle Zufallsvariablen beobachtet werden können. Die erste Veröffentlichung, die auch Anwendungen berücksichtigt, war [DLR77], mit [NH93] ist eine gute zugängliche Einführung gegeben, die auch inkrementelle Varianten des Algorithmus bespricht.

Der EM-Algorithmus hat seinen Namen von den zwei Phasen, in die der Algorithmus unterteilt werden kann: Die Schätzphase (**E**xpectation) bestimmt die unbeobachteten Zufallsvariablen und benutzt dabei gegebene Modellparameter. Die anschließende Maximierungsphase (**M**aximation) bestimmt auf Basis dieser Schätzung der unbeobachteten Zufallsvariablen eine neue Schätzung der Modellparameter. Die Aussage von [DLR77] ist nun, daß eine Iteration dieses Verfahrens tatsächlich die likelihood der Parameter des Gesamtproblems verbessert.

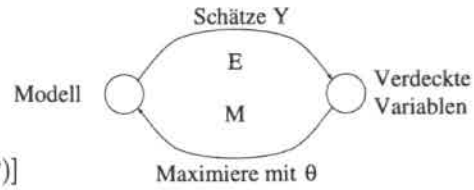
Nehmen wir also an, wir haben eine Zufallsvariabe Z beobachtet, aber nicht Y. Auf Grund dieser Daten wollen wir eine maximum likelihood Schätzung der Parameter θ eines Modelles für Z und Y finden. Das Problem sei nur schwer direkt zu lösen, aber das Problem vereinfache sich stark, wenn Y bekannt wäre. Sei $P(y,z,|\theta)$ die Wahrscheinlichkeitsdichte, bezüglich der die maximum likelihood Schätzungen vorzunehmen sind und es sei eine (gute) Anfangsschätzungen von $\theta^{(0)}$ bekannt. Die Randverteilung $P(z|\theta) = \sum_y P(y,z|\theta)$ sei bekannt. Der EM-Algorithmus generiert daraus eine Folge von Parameterschätzungen $\theta^{(1)}, \theta^{(2)}, \ldots$ durch die iterierte Anwendung der folgenden Schritte

E Schritt

$$\hat{P}^{(t)}(y) = P(y|z,\theta^{(t-1)})$$

M Schritt

$$\theta^{(t)} = argmax_\theta E_{\hat{P}^{(t)}(y)}[\log P(y,z|\theta)]$$



wobei $\hat{P}$ eine Verteilung über Y ist. [DLR77] hat bewiesen, daß auf diese Weise die echte likelihood $L(\theta)$ in jedem Schritt erhöht wird, bis ein lokales Maximum erreicht wird. Es reicht bereits aus, im Maximierungsschritt $E_{\hat{P}^{(t)}(y)}[\log P(y,z|\theta^{(t)})]$ gegenüber $E_{\hat{P}^{(t)}(y)}[\log P(y,z|\theta^{(t-1)})]$ zu erhöhen (generalisiertes EM-Verfahren). In den meisten Anwendungen ist der **E**-Schritt der einfachere, während der **M**-Schritt aufwendig ist. Bei vielen Anwendungen ist es sinnvoll, den **E**-Schritt nur partiell vorzunehmen und nur einen Teil von y neu zu schätzen (siehe [NH93]). Die Verfahren zum Wortalignment nach [BPPM93a] benutzen den EM-Algorithmus und das in dieser Arbeit entwickelte iterierte Verfahren zur Wortklassenbildung erhält durch den EM-Algorithmus eine statistische Deutung (siehe Lemma 3.7).

2.6 Maße für den Zusammenhang von Ereignissen

In dieser Arbeit werden Maße für den Zusammenhang von Ereignissen an unterschiedlichen Stellen verwendet. Diese Fragestellungen tritt vor allem bei folgenden Problemen auf

- Bestimmung von Übersetzungen von Worten (Zusammenhang der Ereignisse, daß in einem bilingualen Corpus zwei Worte in der Übersetzung einander zugeordnet sind)
- Bestimmung von Sequenzengrenzen (Zusammenhang des Ereignisses, daß zwei Wortfolgen im Corpus nacheinander stehen)
- Subkategorisierungsinformation von Verben

Unter den Maßen für den Zusammenhang gibt es

- heuristische Maße
- informationstheoretische Maße
- Maße auf der Basis von Teststatistiken

Alle Zufallsvariablen X sollen im folgenden binär sein und das Vorhandensein eine Wortes x (Ereignis x) bzw. das nicht-Vorhandensein (Ereignis $\neg x$) beschreiben. In der Regel interessieren die absoluten Zahlen des Maßes nicht, es soll nur ein Vergleich zwischen verschiedenen Paaren von Worten (x, y) erstellt werden.

Ein Beispiel für ein rein heuristisches Maß ist die Zusammenhangsdefinition bei [KR93], das sich leicht umgeformt auch als $\frac{p(x,y)}{p(x)+p(y)}$ schreiben läßt. Das wichtigste informationstheoretische Zusammenhangsmaß ist die *mutual information* (siehe Abschnitt 2.4).

Maße auf der Basis der Teststatistik werden für die linguistische Anwendung von [Dun93] vorgeschlagen. Man stellt dazu die Hypothese auf, daß $X|y$ und $X|\neg y$ aus der gleichen Verteilung stammen und kann dazu unterschiedliche Verteilungsannahmen treffen. Aufgrund der in Texten vorkommenden seltenen Ereignisse ist die Binomialverteilung (bzw. die Multinomialverteilung) die einzige adäquate Verteilung. [Dun93] gibt einen *likelihood ratio Test* für dieses Problem an und verwendet die Teststatistik λ direkt als Zusammenhangsmaß. Für die Binomialverteilung ergibt sich das Maß

$$\log(\lambda) = \log L(p, k_1, n_1) + \log L(p, k_2, n_2) - \log L(p_1, k_1, n_1) - \log L(p_2, k_2, n_2)$$

mit

$$\log L(p, k, n) = k \log p + (n - k) \log(1 - p)$$

wobei $p_i = \frac{k_i}{n_i}$, $p = \frac{k_1+k_2}{n_1+n_2}$ und $k_1 = count(x)$,$n_2 = count(x) + count(\neg x)$, $k_2 = count(y)$,$n_2 = count(y) + count(\neg y)$. Unter den oben gennanten Voraussetzungen gilt

$$\log(\lambda)/(n_1 + n_2) = \mathcal{MI}(X, Y)$$

da

$$\begin{aligned}\log(\lambda)/(n_1 + n_2) &= p(y)d(p(X|y)||p(X)) + p(\neg y)d(p(X|\neg y)||p(X)) \\ &= E_p(X, Y)p(X, Y) \log \frac{p(X|Y)}{p(X)}\end{aligned}$$

Kapitel 3

Unüberwachte hierarchische Analyse einsprachiger Corpora

Dieses Kapitel soll vollautomatisierte Techniken beschreiben, die die Analyse eines einsprachigen Korpus ermöglichen. Eine Analyse eines Corpus soll als die Ableitung eines zumindestens grammatikähnlichen Modelles verstanden werden, das den Corpus in gewisser Weise adäquat beschreibt. Eine wesentliche zusätzliche Schwierigkeit besteht darin, daß hierfür nur ein kleiner, spontansprachlicher Corpus zur Verfügung steht, auf dem statistische Analysen nur unter hohem Aufwand sinnvoll sind[1].
Zur Analyse werden im wesentlichen zwei Operatoren verwendet

Klasseneinteilung Den elementaren Elementen eines Corpus werden Klassen zugeordnet (siehe Abschnitt 3.1).

Verkettung Ketten von Elementklassen werden verkettet und durch ein neues Symbol repräsentiert (siehe Abschnitt 3.2).

Diese Operatoren lassen sich iteriert anwenden und führen zu einer schrittweisen Analyse des Corpus (siehe Abschnitt 3.3).

Die Qualität solcher Analysen kann zum einen dadurch bestimmt werden, daß man sich die erzeugten Sequenzen anschaut und sich von der Plausibilität der Analyse überzeugt. Zu diesem Zweck werden in dieser Arbeit verschiedene Visualisierungstechniken beispielhaft vorgeführt, die im Rahmen der Experimente benutzt wurden. Zum anderen soll ein besser objektivierbares komplexitätstheoretisches Maß vorgestellt werden, das die Güte der Modellbildung beschreibt (siehe Abschnitt 3.3.2). Dieses Maß beschreibt im Prinzip die Güte eines Kompressionsverfahrens auf der Basis der hier vorgestellten Methoden.

Mit dem in der Spracherkennung üblichen Maß der Perplexität wird nachgewiesen, daß dieses Sprachmodell wesentlich adäquater ist als die oft verwendeten n-gramm Modelle, die keine Struktur erzeugen. Weitere Experimente zur Bestimmung von Sprechakten runden dieses Kapitel ab.

[1] Vergleichbare Analysen zur Distributionsanalyse auf Worten wie [FC92] [Fin93] gehen von Corpora mit einigen zehn Millionen Worten aus, der spontansprachliche Corpus besteht nur aus einigen zehntausend Worten. Alternative Ansätze zur Sequenzensuche benutzen ebenfalls kleine, aber nicht spontansprachliche Corpora und man darf erwarten, daß in einem spontansprachen Corpus gerade die Sequenzeninformation stark verrauscht ist.

3.1 Distributionsanalyse

Die Aufgabe dieses Abschnitts ist es, linguistisches Wissen über sinnvolle Klasseneinteilungen für statistische Verfahren nutzbar zu machen. Mit [Fin93] ist eine hervorragende Abhandlung über die verschiedenen in der Linguistik entwickelten Ansätze gegeben, so daß sich diese Kapitel ausschließlich mit der hier verwendeten Distributionsanalyse beschäftigt. Vergleichbare Ansätze werden auch in [BdP+92] [KN93] verwendet, um klassenbasierte n-gram Sprachmodelle zu erstellen. Der Vorteil des hier beschriebenen Ansatzes ist die gute praktische Durchführbarkeit, die sowohl durch die Übertragung bekannter statistischer Verfahren mit definierten Eigenschaften und der damit hohen Qualität der Klassifikation als auch durch den vergleichsweise geringen Bedarf an Rechenzeit zu erklären sind. Darüberhinaus scheinen die nach einem manuellen Vergleich die gefundenen Wortklassen eher linguistischen Kriterien zu genügen als die auf dem gleichen Corpus angewandeten Verfahren aus [BdP+92][KN93].

Das verwendete linguistische Kriterium ist der Ersetzbarkeitstest, der in [Fin93, Definition 3.1.1, Seite 44] wie folgt zitiert ist

Definition 3.1 (Ersetzbarkeitstest) *Hat ein Wort oder eine Phrase die gleiche Verteilung wie (d.h. sie kann immer ersetzt werden durch) ein Wort oder eine Phrase eines unbekannten Typs, dann haben beide den gleichen Typ.*

Der Ersetzbarkeitstest ist ein wesentliches Werkzeug der strukturalen Linguistik, das trotz der Hinwendung zum transformationellen Ansatz weiterhin allgemein akzeptiert wird. Dieses Kriterium ist noch nicht einer statistischen Analyse zugänglich und wird in [Fin93, Definition 3.2.1, Seite 53] erweitert

Definition 3.2 (Statistischer Ersetzbarkeitstest) *Treten ein Wort bzw. eine Phrase in ähnlichen Kontexten wie ein anderes Wort bzw. eine andere Phrase auf, dann gebe ihnen eine ähnliche Kategorie.*

Ich möchte diese Definition noch etwas radikaler formulieren als

Definition 3.3 *Ein Wort bzw. eine Phrase wird durch die Kontexte (die wiederum aus Worten bzw. Phrasen bestehen), in denen sie auftritt, repräsentiert.*

Diese Definition ist in einer gewissen Weise zyklisch: Wenn man weiß, wie der Kontext eines Wortes zu repräsentieren ist, dann kennt man eine Repräsentation der Worte. Um aber den Kontext eines Wortes adäquat bestimmen zu können, ist die Kenntnis einer Repräsentation der Worte nötig.

Initiale Klasseneinteilung und Zipfsche Regel Im folgenden gehen wir immer davon aus, das es für jedes Wort bereits eine Repräsentation gibt und diese Repräsentation als Zugehörigkeit zu einer bestimmten Wortklasse angegeben wird. Zu Beginn der Analyse, bei der noch keine Klassifikation vorliegt, muß man eine sinnvolle ad-hoc Einteilung wählen. Aus statistischen Gründen wird man bei n-Wortklassen, die vorgegeben sind, die $n-1$ häufigsten Worte jeweils einer einelementige Klasse zuordnen und die restlichen Worte in eine Klasse zusammenfassen. Danach kann man das Verfahren iterieren und die jeweils letzte Wortklassifikation benutzen, um die folgende zu schätzen. [Fin93] benutzt ausschließlich die ad-hoc Klasseneinteilungen und iteriert das Verfahren nicht. Seine Begründung für diese Wortklasseneinteilung ist, daß in den häufigsten Worten die sichersten statistischen Informationen zu finden sind. [Fin93, Seite 12ff] wie auch [Bri93, Seite 41ff] gehen davon aus, daß die Häufigkeitsverteilung in natürlichsprachlichen Texten im wesentlichen der *Zipfschen Regel* unterliegen, die starke Ungleichverteilungen vorhersagt:

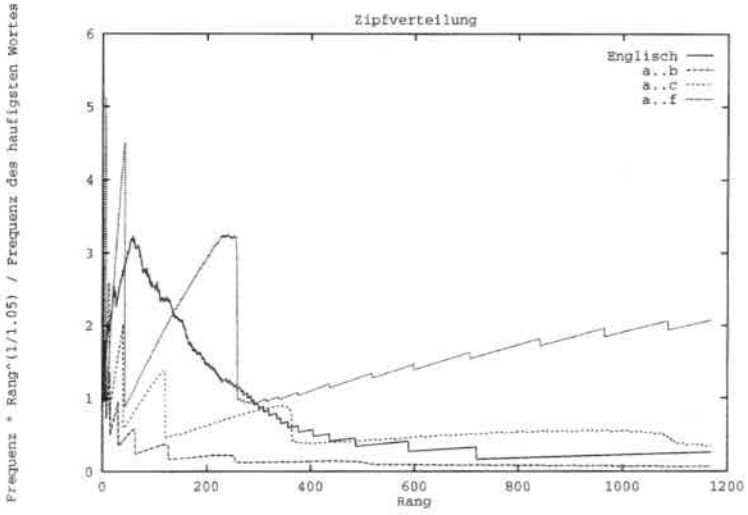


Abbildung 3.1: **Zipfsche Verteilung:** Diese Abbildung zeichnet Zipf-Kurven für den englischen Trainigscorpus im Vergleich zum Affen mit einer Schreibmaschine mit einer Untermenge der Buchstaben und der Leertaste. Die Kurven wären ideale Zipfkurven, wenn sie parallel zur x-Achse in der Höhe 1 verliefen. Die Kurve für den englischen Corpus sieht in einem Anfangsbereich sehr viel günstiger als durch die Zipfverteilung zu erwarten wäre und ist bei hohen Rängen deutlich ungünstiger. Auffällig ist die Glätte der Verteilung. Die Abweichung von der idealen Zipfkurve ist in ihrer Größe geringer als die des Affens. Insbesondere scheint die englische Kurve mit der eines Affen mit etwa 2.5 Buchstaben und der Leertaste vergleichbar zu sein (dies entspricht einer Entropie pro Buchstabe von $\log_2(2.5+1) = 1.8$). Dies deckt sich ungefähr mit Vermutungen, daß die Entropie des Englischen unter 1.75 bit pro Buchstabe liegt [BPPM93b], die auf großen Corpora bestimmt wurde. Das Skript *monkey* erzeugt einen Zufallscorpus und das Skript *freq -zipf* erzeugt die Ausgaben für *gnuplot*.

Nach der durch Mandelbrot verbesserten Schätzung gilt, daß die Wahrscheinlichkeit des m-ten Wortes in der Rangfolge der Wahrscheinlichkeiten proportional zu $m^{-1,05}$ ist [Fin93, Seite 12]. Damit ist nur die Schätzung auf der Basis eines kleinen Teiles der im Corpus vorhanden Worte sinnvoll, da alle anderen Worte zu selten bzw. gar nicht vorkommen: Bei einem Vokabular von 50000 Worten decken die häufigsten 100 Worte bereits $\frac{\log 100}{\log 50000} = 42\%$ des Corpus ab, die häufigsten 2000 Worte bereits 70%.

Der Zipfschen Regel liegt das Modell eines sehr einfachen stochastischen Prozess zugrunde, der in der Einführung durch Georg Miller in [Zip35] wie folgt beschrieben ist (zitiert nach [Bri93, Seite 41f]):

> Suppose that we acquired a dozen monkeys and chained them to typewriters until they had produced some very long and random sequence of characters. Suppose further, that we define a "word" in this monkey-text as any sequence of characters between successive spaces. And suppose finally that we counted the occurences of these "words" in just the way Zipf and others counted the occurences of real words in meaningful texts. When we plot our results in the same manner, we will find exactly the same "Zipf curves" for monkeys as for human authors.

Die so vorhergesagte Ungleichverteilung in den Häufigkeitsverteilungen von Worten könnte zu der Schlußfolgerung führen, daß auch bei einer Ausdehnung der Trainingsmenge nie genügend Daten zur Schätzung der Parameter eines statistischen Modelles von Sprache zur Verfügung steht. Die Konsequenz daraus muß sein, daß durch die Aufdeckung von Gemeinsamkeiten zwischen Worten diese Verteilung in einen Bereich gebracht werden muß, in dem zuverlässige Schätzungen möglich sind. Eine Methode dies zu tun, ist die hier vorgestellte Distributionsanalyse. Insbesondere durch das iterierte Klassifikationsverfahren wird die Sicherheit bei der Schätzung verbessert, da dadurch auch die Distributionen sicherer aus dem Datenmaterial zu schätzen sind.

Formalisierung Ähnlich wie [Fin93, Kapitel 5.1], soll jetzt definiert werden, was genau der Kontext eines Wortes ist. Die Definition von [Fin93] basiert auf Textströmen, die mit bestimmten Operatoren kombiniert werden können. Wenn man aber betrachtet, welche Definitionen tatsächlich ausgeschöpft werden, stellt man fest, daß im wesentlichen immer der Ausgangscorpus betrachtet und zusätzlich zu jedem Wort seine jeweilige Klasse eingetragen wird. Diese Definition ist in gewisser Weise allgemeiner, da sie nicht nur das klassische Zählen der Worte im Corpus gestattet.

Definition 3.4 (Kontextrepräsentation) *Sei $w_1 \cdots w_n$ eine Sequenz von Worten, dann ist der Kontext $[i_{low}, i_{high}]$ an der Stelle j die Multimenge von Worten $\langle w_{j+i_{low}}, \ldots, w_{j+i_{high}} \rangle$. Der Kontext eines Wortes w bezüglich der Sequenz $w_1 \cdots w_n$ und des Intervalls $[i_{low}, i_{high}]$ ist die Vereinigung der Kontexte an den Stellen j mit $w = w_j$.*

Sei $t : \langle w_1 \cdots w_n \rangle \mapsto \mathcal{R}^m$ eine Vektorrepräsentation *und $f^k : \prod_{i=1}^{k} \mathcal{R}^m \mapsto \mathcal{R}^m$ eine* Kontextzusammenfassung. *Die Vektorrepräsentation des Wortes w bezüglich des Intervalls $[i_{low}, i_{high}]$ ist dann $f^k(\langle t(w^1), \ldots, t(w^k) \rangle)$, wobei $\langle w^1, \ldots, w^k \rangle$ der Kontext von w bezüglich $[i_{low}, i_{high}]$ ist. Die Vektorrepräsentation des Wortes w bezüglich der Intervalle $[i_{low_1}, i_{high_1}], \ldots, [i_{low_c}, i_{high_c}]$ wird aus den Vektorrepräsentation der einzelnen Intervalle durch Aneinanderreihung der Vektoren hergestellt.*

Beobachtung 3.5 *Eine besonders einfache Varianten der Vektorrepräsentation entsteht, wenn $t(w) = (0, \ldots, 0, 1, 0, \ldots, 0)$ und f^k die Addition der Vektoren in der Multimenge darstellt: t stellt dann eine Klasseneinteilung dar und f^k zählt, wie oft eine Wortklasse im Kontext dieses Wortes vorgekommen ist.*

Ein anderer interessanter Spezialfall ist, daß t die Zugehörigkeit des Wortes w zu den Klassen einer Klassifikation der Worte in fuzzy logic darstellt. f^k kann dann als Disjunktion über alle Vektoren der Multimenge definiert werden.

In beiden Fällen kann man ein neues t aus einer Vektorrepräsentation eines Wortes bezüglich eines Kontextes ableiten, indem man ein Clusterverfahren benutzt.

Aus diesen Überlegung ergibt sich ein iteratives Verfahren, das die Vektorrepräsentationen der Worte und damit auch die der (fuzzy) Klasseneinteilung schrittweise verbessert und durch das Skript *cluster* einfach anwendbar ist:

Algorithmus 3.6 (Iterative Bestimmung von Wortrepräsentationen)

1. *Schätze eine Anfangsklassifikation der Worte und initialisiere t entsprechend.*
2. *Wiederhole die Schritte 3-5, bis die Repräsentation genau genug ist.*
3. *Bestimme aus t, einer Intervallspezifikation und f^k eine neue Vektorrepräsentation für alle Worte im Corpus.*

Repräs.	Elemente	Repräs.	Elemente
couple	more couple another	have	be got have
do	do get	'bout	'bout about
hours	hours weeks	or	or
like	like	that	that those
eighth	fifth first fourth ninth second eighth sixth thirtieth sixteenth seventh third	monday's	rather thursday's you'll you'd prefer wednesday's there's that'd monday's friday's
early	late early sometime	february	october august may november february july june april march december
+h#+	$1 +nonhum+ +h#+ +ls+ +uh+ +um+ +human+	probably	probably it's
for	for with	is	is are
can	can could should let's we'll	meeting	meeting time
well	okay well yeah	afraid	afraid yes unfortunately sure sorry
my	my your	mornings	afternoons schedule mornings
our	our	if	if would
any	any some other	before	before around after
i'm	i'm	the	the next twenty
open	available free open	bye	by bye let why thanks
not	not really	+muell+	+muell+ +paper+
'till	most 'till till later during starting	morning	day week morning afternoon
want	set try make need plan take want wanna	ten	am ten nine noon eight
was	was only	sounds	sounds that's looks
anytime	again anytime except either	+ah+	+ah+
we	i we you	over	over
sunday	sunday saturday	else	today else better said
it	it	days	days times
to	to	together	go something back anything together
up	up town	you're	you're we're
no	no actually	then	then +click+
twentieth	nineteenth fifteenth $2 thirteenth twentieth weekend fourteenth eighteenth office tenth eleventh seventeenth	totally	completely filled leaving full gone going kinda tied very were definitely still getting looking totally
bad	bad good	all	all
a	a	an	an hour
busy	booked out busy	how	how what
friday	thursday monday wednesday friday tuesday	here	here
as	as	me	me
'cause	'cause because when	say	say meet
on	at in on	see	see
seminars	meetings class seminar vacation classes seminars	entire	f end best entire last rest whole
i'll	i'll i've	lunch	lunch
this	this	look	look work
minute	ice month now six way possibility perfect enough away evening cream idea possible minute things lecture twelfth place stuff though	put	us eat fit put able call come reach find arrange give hate continue keep squeeze mind talk tell wait leave discuss start
how's	what's does how's sound	just	just
fact	perhaps also both fact even much coming possibly than following which	might	can't doesn't i'd gonna wouldn't that'll will won't might
o'clock	pm five o'clock thirty	off	off done right
of	of through	maybe	maybe
there	there	great	fine great alright
and	so and but	guess	know guess don't think
from	until from between	eleven	one two four eleven twelve three
too	too pretty		

Abbildung 3.2: **Wortklassifikation auf 100 Klassen:** Die Klassifikation wurde auf dem englischen Terminabsprache Task abgeleitet, der eine Länge von 44609 Worten hat. Das Verfahren wurde achtfach iteriert, als Kontext wurde $[-3,-1][-1,-1][1,1][1,3]$ gewählt. Diese Tabelle zeigt alle Klassen mit höchstens 30 Worten.

4. *Berechne eine (fuzzy) Klasseneinteilung der Worte auf der Basis der im vorherigen Schritt bestimmten Klasseneinteilung durch ein entsprechendes Clusterverfahren.*

5. *Bestimme ein neues t entsprechend dieser Klasseneinteilung.*

Ähnliche schrittweise Verfahren sind in der Statistik als EM-Algorithmen bekannt (siehe Abschnitt 2.5). Um den Zusammenhang formalisieren zu können, wird ein Modell aufgestellt, das auf einer Klassenbildung basiert und dessen Modellparameter mit einem EM Algorithmus geschätzt werden können, der im wesentlichen dem obigen Algorithmus entspricht.

Lemma 3.7 (statistisches Modell der Wortklassifikation)
Sei t_θ abhängig von den Modellparametern θ gegeben und f^k sowie die Intervallspezifikation I fest aber beliebig. Die sich daraus ergebende Vektorrepräsentation eines Wortes w bezügliches des Corpus sei dann als $t'_\theta(w)$ gegeben. Sei ein Algorithmus $\mathcal{A}$ gegeben, der bezüglich des Corpus

$$\theta = argmax_{\theta'} E[\log P(t'_{\theta^*}(w)|\theta'))]$$

$$\text{mit } P(t'_{\theta^*}(w)) = \sum_{t'_{\theta^*}(w')=t'_{\theta^*}(w)} P(w')$$

berechnet.
Dann berechnet folgender Algorithmus schrittweise bessere Schätzungen (sofern noch kein lokales Maximum erreicht ist) für $\theta = argmax_{\theta'} E[\log P(t'_{\theta'}(w)|\theta'))]$ und stellt einen EM Algorithmus dar:

1. *Wähle θ beliebig*

2. *Berechne mit $\mathcal{A}$ das neue θ zu $argmax_{\theta'} E[\log P(t'_\theta(w)|\theta'))]$ mit $P(t'_\theta(w))$ wie oben*

3. *Gehe zu 2*

Der Beweis ist trivial, wenn man $t'_\theta(w)$ mit der Zufallsvariablen y in der Formulierung des EM-Algorithmus (siehe Abschnitt 2.5) indentifiziert. Unter der Annahme eine multimodalen Verteilung auf den Vektorrepräsentationen lassen sich die aus dem Kapitel 2.1 bekannten Verfahren zur Klassifikation benutzen. Als Abstandsmaß hat sich in dieser Arbeit, ebenso wie in [Fin93], der euklidische Abstand der Ränge bewährt. Beispiele für Wortklassifikation und abgeleitete Größen finden sich in den Abbildungen 3.2, 3.3, 3.4, 3.5 und 3.6.

Beobachtung 3.8 (Informationstheoretische Modelle) *In [BdP+92] wird ein Methode eingeführt, eine Klasseneinteilung von Worten vorzunehmen, die ein Bigramm-Sprachmodell in gewisser Weise optimiert. Aus [BdP+92] läßt sich ohne weiteres ableiten, daß dies äquivalent dazu ist, die Likelihood $E[\log(Pr(c_1|c_2))] = H(C_1|C_2)$ zu minimieren, wobei c_1 eine Wortklasse ist, die im Corpus auf das Wort c_2 folgt (ebenso die Zufallsvariablen C_1 und C_2).*

Wir kodieren in die Modellparameter θ eine klassische Klasseneinteilung und in t die entsprechende Vektorrepräsentation des Wortes. f^k summiere alle Vektoren und teile den resultierenden Vektor durch die Summe seiner Elemente. Sei $t'(w)$ die so resultierende Vektorrepräsentation bezüglich des Intervalls $[1,1]$, dann ist $t'(w)$ die Maximum-Likelihood Schätzung für $(Pr(c_1|w), \ldots, Pr(c_n|w))$, wobei $c_1, \ldots, c_n$ die durch t induzierte Benennung der Klassen ist. Damit können die Vektorrepräsentationen des Kontextes der Worte als Wahrscheinlichkeitsverteilung über den Klassen gedeutet werden.

Ein informationstheoretisches Maß über solchen Vektoren läßt sich dann auf Wortklassen ausdehnen, wenn man man einen Algorithmus hat, der die die logarithmierte Likelihood

$$L = \log \prod_{i=1}^{n} \prod_{j=1}^{n'} Pr(c_i|c'_j)^{Pr(c_i,c'_j)}$$

maximiert, wobei c'_j die neuen Wortklassen sind. Man beachte, daß diese Formel gerade $\mathcal{H}(C|C')$ darstellt, wobei C und C' die entsprechenden Zufallsvariablen sind. Es genügt also ein Algorithmus zu kennen, der die bedingte Entropie der folgende Klasse bei der Zusammenfassung der Worte zu Klassen minimiert. Wenn man dieses Verfahren iteriert anwendet, kann man also ein klassenbasiertes Bigramm schätzen.

Bemerkung 3.9 (Technische Realisierung des iterierten Verfahrens)
Im Kapitel 2.1.2 wurde bei der Vorstellung der Initialisierungsverfahren für das Hill-Climbing-Verfahren erwähnt, daß die Initialisierung mit bekannten Werten *in der Regel das beste Verfahren darstellt. Bei der iterierten Wortklassifikation ist eine gute Anfangsschätzung der Wortklassifikation möglich, nämlich die Wortklassifikation, die in der letzten Iteration abgeleitet wurde. Der Einsatz dieses Verfahrens hat sowohl die Rechenzeit erheblich verkürzt als auch die Präzision der Klassifikation. Falls eine Wortklassifikation mit sehr vielen Klassen gewünscht wird, ist es nicht sinnvoll, das iterierte Verfahren mit diesen vielen Klassen durchzuführen, da die resultierenden Kontextrepräsentationen zu ungenau sind. Stattdessen kann man in den Zwischenschritten mit einer geringeren Klassenzahl arbeiten und erst die letzte Klassifikation auf der gewünschten Klassenzahl berechnen. Die Initialisierung kann auch im letzten Schritt mit bekannten Werten erfolgen, es muß lediglich in einer Vorverarbeitungsphase die alte Klassifikation verfeinert werden. Das gesamte Verfahren wie oben beschrieben ist in dem* PERL *-Skript* cluster *integriert, daß dazu neben anderen Programmen das Clusterverfahren* ccluster *und das Programm* distr *zur Berechnung der Kontextrepräsentation benutzt.*

Repräs.	Elemente	Repräs.	Elemente
D	B D F G SCH	A	A I O U
CH	N CH NG	AEH	E AI AU EU AEH UEH
R	K P R	UE	AE OE UE OEH
T	L M S T	IE	AH EH ER IE OH UH
H	H J V Z	SIL	+K QK +EH +H# +GH +GN SIL

Abbildung 3.3: **Klassifikation von Phonemen:** Dieses Beispiel zeigt die Klassifikation in 100 Klassen bei Verwendung eines Kontextes $[-1,-1],[1,1]$ auf einer phonetischenn Transkription von englischen Terminabsprachedialogen. Das Ergebnis zeigt, daß die gefundenen Klassen zum Teil auch akustische Ähnlichkeiten modellieren. Dies ist nicht verwunderlich, da der Vokaltrakt des Menschen in einer natürlichen Sprache nur bestimmte Folgen von Stellungen, die besonders einfach zu sprechen sind, durchläuft.

Größe	Repräs.	Elemente
		Can
25	we	i if we can i'd i'm why could i'll i've you're it's gonna sounds don't what when will let's we'll we're looks would
50	could	we can you could should don't let's we'll
100	can	can could should let's we'll
200	can	can could
		Town
25	arrange	us eat fit set try back call case come else find arrange give co ntinue into keep squeeze make mind plan schedule take talk town leave discuss
50	lecture	tennis meetings class away seminar vacation town classes lecture seminars
100	up	up town
200	town	town
		Free
25	available	up booked meetings all any bad off out too available busy done better fine free full gone good right late look coming open pretty looking
50	gone	booked out available busy free full gone open tied
100	open	available free open
200	free	free
		Saturday
25	february	sunday october august may the november february both around jul y june last saturday next early following april whole march either december twen ty those
50	wednesday	thursday monday saturday wednesday friday afternoon tuesday
100	sunday	sunday saturday
200	saturday	saturday
		Time
25	+um+	$1 is or so thursday and but how +nonhum+ here monday wednesday that then time again well +muell+ friday +paper+ +ah+ +h#+ sometime +ls+ +uh+ +um+ maybe +click+ there +human+ tuesday
50	day	day days weekend time week hours weeks times
100	meeting	meeting time
200	time	time
		To
25	for	at in of on to before for through until from over than till after wi th anytime between during
50	for	to for with
100	to	to
200	to	to
		Sorry
25	tomorrow	as by today bye now 'cause enough long much because tomorrow th anks where sorry though tight unless
50	yes	no yes unfortunately sure well yeah sorry actually
100	afraid	afraid yes unfortunately sure sorry
200	afraid	afraid yes 'cause unfortunately fact sorry
		Perhaps
25	perhaps	f somewhere $2 including fifteenth bet boy what's neither beside s perhaps filled does down each entire fact otherwise even goes anyway most pick although vacation possibly possible rest same earlier later there's classes how 's within anywhere mention which solid except january starting stuff seminars ev ery conference
50	perhaps	as oh bye now 'cause perhaps although because how's which though unless
100	fact	perhaps also both fact even much coming possibly than following wh ich
200	same	perhaps even most coming possibly same later following
		I'll
25	we	i if we can i'd i'm why could i'll i've you're it's gonna sounds don' t what when will let's we'll we're looks would
50	i	i i'm i'll i've
100	i'll	i'll i've
200	i'll	i'll

Abbildung 3.4: **Klasseneinteilung ausgewählter Worte:** Für jedes ausgewählte Wort werden für Klasseneinteilungen verschiedener Anzahl die zugehörigen Klassen dargestellt. Der Klassifikationsvorgang wurde achtfach iteriert, der Repräsentant eines Clusters ist das Wort, dessen Distribution dem Klassenmittelpunkt am nächsten liegt. Als Kontext wurde $[-3,-1],[1,1],[1,1],[1,3]$ gewählt.

for to on from	then how	or +h#+ and	+ah+ see	well sounds it no	not like guess	have do just
'bout	+muell+		great			probably can if we i'll
my the a this of	as	me bad		afraid 'cause	was	you're might
any all	open up busy too		february	how's fact		want together say look
that is i'm			seminars	twentieth		put
days	early before anytime	'till	mornings entire	totally	bye monday's	
eighth morning		couple over off	our			super
	an	preferenc			reserve	ha
sunday friday	meeting maybe there	else minute		dr		
ten lunch o'clock eleven	hours here		wedding	lost	stupendou	who's

Abbildung 3.5: **Visualisierung der Klassenzentren:** Die durch die Klassenbildung aus 3.2 entstanden Klassenmittelpunkte wurden mit einer selbstorganisierenden Karte nach Kohonen mit 7 mal 10 Neuronen in den zweidimensionalen Raum projeziert. Die einzelnen Bereiche der Karte geben deutlich unterschiedliche Konzepte wieder. So finden sich z.B. im linken unteren Bereich zeitliche und örtliche Angaben. Das Skript *cluster* erzeugt diese LaTeX-Ausgabe bei Angabe der Option *-vis*, die Klassifikation wird durch das Paket *SOM-PAK* von der Universität Helsinki berechnet.

Abbildung 3.6: **Fuzzy Klassifikation:** Die Abbildung zeigt die Klassifikation der in Abbildung 3.4 selektierten Worte bei 25 Klassen und $m = 1.2$, $w = 6$. Man beachte auch die Ähnlichkeit von *i'll* und *can*, aber auch die von *saturday* und *time* sowie die schlechte Klassifikation von *town*. Ein voll ausgefülltes Kästchen wie bei *can* steht für eine Zugörigkeit von 1. Die Fuzzy Klassifikation bietet sich als Vorverarbeitung für ein neuronales Netz an.

3.2 Sequenzenanalyse

Der zweite Operator, der hier betrachtet werden soll, dient der Verkettung von Worten bzw. beliebigen anderen Einheiten zu größeren Einheiten. [MM91][BMS90] ist die derzeit einzige Arbeit, die eine solche Analyse auf natürlichsprachlichen Texten versucht hat.

In anderen Bereichen der statistischen Sprachverarbeitungen wird ebenfalls mit Methoden gearbeitet, die Sequenzen auffinden können, obwohl das Ziel dieser Arbeiten nicht explizit das Auffinden von Sequenzen war.

- [SW94] hat versucht, die Perplexität eines Sprachmodelles durch eine Zusammenfassung häufig auftretender Sequenzen zu verringern
- [WW94] hat ein Verfahren entwickelt, das aktuelle Dialogzustände mit einem Hidden Markov Modell schätzt und stellte fest, daß die Unterteilung einer Äußerung an den Wechseln des Dialogzustandes ein gutes Kriterium für Sequenzengrenzen ist

Das Suchen von Sequenzen hat eine enge Verwandschaft mit einem bottom-up Verfahren zur Zerteilung von Texten, bei dem im Prinzip nichts anderes geschieht als Sequenzen zu suchen, den gefundenen Sequenzen einen Namen zu geben und anschließend den Vorgang zu wiederholen. Man kann also alle Verfahren zur automatischen Induktion von Grammatiken als Verfahren zur Sequenzenanalyse interpretieren.

In der Arbeit von [Fin93] werden als Sequenzenkandidaten ausschließlich Sequenzen der Länge 2 und 3 verwendet, die häufig im Text vorkommen. Dieses Kriterium muß als zu schwach gelten, da es auch bei iterierter Anwendung längere Floskeln nicht genügend berücksichtigt.

Ziel der Sequenzenanalyse war es in dieser Arbeit zunächst, Floskeln und Redewendungen zu identifizieren. Dabei stellte sich jedoch rasch heraus, daß diese Floskeln und Redewendungen selbst in einem spontansprachlichen Corpus deutlich von grammatikalischen Regularitäten überlagert werden. Der ursprüngliche Plan, diese Floskeln und Redewendungen zusammenzufassen, um das Alignment zu verbessern, mußte somit aufgegeben werden und wurde durch die Zielsetzung, die Struktur der Sätze möglichst genau zu schätzen, ersetzt.
Die Suche nach Sequenzen geschieht im wesentlichen in zwei Schritten:

1. Bestimme die Güte einer Wortfolge als Kandidat für eine Sequenz.
2. Bestimme die tatsächlich im Text vorhandenen Sequenzen auf Grundlage dieses Maßes.

Bei der Entwicklung von Algorithmen zur Sequenzensuche war es hilfreich, zunächst Buchstaben zu Worten und Morphemen zusammenzufassen, da dies zum einen erheblich schwerer zu sein scheint als die Ableitung einer syntaktischen Struktur aus Wortfolgen und zum anderen die Ergebnisse sehr viel besser interpretierbar sind.

3.2.1 Kriterien für die Güte einer Sequenz

Die Maße für die Güte einer Sequenz lassen sich im wesentlichen in folgende Gruppen einteilen

- direkte Maße
- indirekte Maße, die aus Zusammenhangsmaßen entwickelt werden (siehe Abschnitt 2.6)

Alle Maße bis auf die (iterierte) Markierungsfrequenz sind mit den zwei unten beschriebenen Selektionsalgorithmen in dem Programm *pro* integriert. Das PERL -Skript *chunker* kann die Ausgaben von *pro* eigenständig zur Sequenzenselektion benutzen und wird von *build_corpus* benutzt, um die (iterierte) Markierungsfrequenz als Maß zu implementieren.

Direkte Maße Das einzige im Rahmen der Literaturarbeit gefundene brauchbare direkte Maß ist die "innere Stabilität" nach [Suh73] [2], die auch als *Suhotin's Maß* oder *Interior* bezeichnet werden soll. Dieses Maß, das ursprünglich für das Auffinden von Morphemen gedacht war, läßt sich in ähnlicher Weise für die Bewertung der Güte von beliebigen Sequenzen einsetzen. Es ist definiert als

$$g(w_1 \ldots, w_n) = \frac{1}{2n} \sum_{i=1}^{n-1} Pr(w_1 \ldots, w_n | w_1, \ldots w_j \vee w_{j+1}, \ldots w_n)$$

was sich als Maximum-Likelihood Schätzung aus einem Corpus auch wie in der Originalarbeit schreiben läßt

$$g(w_1 \ldots, w_n) = \frac{1}{2n} \sum_{i=1}^{n-1} \frac{Count(w_1 \ldots, w_n)}{Count(w_1, \ldots, w_j)} + \frac{Count(w_1 \ldots, w_n)}{Count(w_{j+1}, \ldots, w_n)}$$

Man kann diese Zahl auch interpretieren als den Erwartungswert der Wahrscheinlichkeit bezüglich aller Prefixe und Suffixe dieser Sequenz, daß diese Prefixe bzw. Suffixe an einer Stelle im Corpus tatsächlich am Ende bzw. am Anfang dieser Sequenz stehen. Wenn diese Zahl hoch ist, muß man davon ausgehen, daß die Sequenz stabil ist.

Ein neues direktes Maß für die Sequenzengüte ist, wie oft eine Folge durch einen Sequenzenmarkierungsalgorithmus im Text gefunden wurde und soll hier als *Markierungsfrequenz* bezeichnet werden. Man kann, sofern man einen Sequenzenmarkierungsalgorithmus hat, der abhängig von einem Sequenzengütemaß arbeitet, dieses direkte Maß iteriert benutzen und in jeder Iteration die Anzahl der benutzt Sequenzen verringern. Auf diese Weise profitieren Sequenzengütemaß und Sequenzenmarkierungsalgorithmus wechselseitig voneinander. Man beachte insbesondere, daß dieses Verfahren andere Werte bildet als das reine Zählen von Vorkommen im Corpus – dieses Maß ist zu stark durch "falsche" Sequenzen verrauscht. Man kann den Einfluß des Ausgangsmaßes in Abbildung 3.11 betrachten.

Algorithmus 3.10 (iterierte Markierungsfrequenz) *Das Verfahren nimmt als Eingabe eine gewünschte Anzahl von Sequenzen, eine Reduktionsrate p und eine nach Güte geordnete Liste von Sequenzen. Die Ausgabe ist eine Liste von Sequenzen mit der gewünschten Anzahl. Das Verfahren führt iterativ folgende Schritte durch und verbessert dabei die Liste schrittweise:*

1. *Falls die Liste der Sequenzen weniger ist als p× gewünschte Anzahl an Sequenzen enthält: Schneide die überflüssigen Sequenzen ab und* **STOP**.
2. *Markiere aufgrund dieser Liste mit einem Sequenzenmarkierungsalgorithmus die im Corpus vorkommenden Sequenzen.*
3. *Falls weniger als die gewünschte Anzahl an Sequenzen im Corpus vorkommen: Schneide die überflüssigen Sequenzen aus der Liste und* **STOP**.

[2] Die Arbeiten von Suhotin wurden in den 50'er oder 60'er Jahren auf Russisch veröffentlicht und [Suh73] ist neben dem daraus entwickelten [Guy91b] die einzig zugängliche Quelle, die diese Arbeiten in einer westeuropäischen Sprache zugänglich macht. Sie sind vor allem wegen ihrer Verbindung von syntaktischen mit statistischen Modellen eine wichtige Quelle.

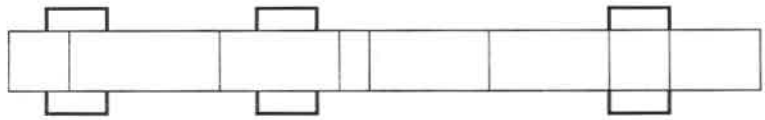


Abbildung 3.7: **Grundidee der Sequenzensuche mit indirekten Maßen:** Es wird angenommen, daß sich der Text in Sequenzen unterteilen läßt und man mit einem Fenster über die Unterteilung fährt. Wenn man das Fenster nicht genau auf eine Sequenz gelegt hat, bestehen die Worte aus einem Suffix der einen Sequenz und einem Prefix der anderen. Man nimmt an, daß diese beiden Teil der Sequenz im Sinne des Maßes, das man übertragen will, nur schwach zusammenhängen. Ein Problem dieser Maße ist es, daß Teilstücke von Sequenzen ebenfalls einen hohen Wert bekommt.

4. *Erstelle eine neue Liste von Sequenzen, indem die markierten Sequenzen nach ihrer Häufigkeit geordnet werden.*

5. *Kürze die Liste so, daß sie höchstens $\frac{1}{p}$ mal so viele Sequenzen enthält wie die vorherige Liste und gehe nach 1.*

Indirekte Maße Allen indirekten Maßen liegt folgende Überlegung zugrunde: Eine Sequenz, die eigentlich keine Einheit darstellt, tritt dann zufällig im Corpus auf, wenn sie aus dem Anfang bzw. Ende einer echten zusammengesetzt wurde (siehe Abbildung 3.7). Dazu werden bekannte Zusammenhangsmaße genutzt:

Definition 3.11 (Indirektes Sequenzengütemaß) *Sei $s(X,Y) \geq 0$ ein Maß, das den Zusammenhang der Zufallsvariablen X und Y bestimmt und bei stärkerem Zusammenhang zwischen den Variablen zunimmt. Dann ist*

$$g_s(w_1,\ldots,w_n) = min_{0<j\leq n} s((x_1,\ldots,x_j),(x_{j+1},\ldots,x_n))$$

das aus s abgeleitete Maß der Sequenzengüte.

Das indirekte Maß, das sich im Rahmen dieser Arbeit zur Sequenzensuche mit Abstand am besten bewährt hat, basiert auf der *mutual information*. Das Verfahren von [MM91][BMS90] ist mit diesem Maß in Grenzen vergleichbar (siehe Absatz *Vergleich zum Verfahren von Marcus* unten). Durch die Anwendung des Maßes der iterierten Markierungsfrequenz ist eine deutliche Verbesserung der Resultate möglich. Dies wurde sowohl durch manuelle Beurteilung als auch durch das MDL Maß (siehe Abbildung 3.8) bestätigt.

Selektion von Sequenzen Die Bestimmung der Güte einer Sequenz läßt noch Spielraum für die tatsächliche Auswahl der Sequenzen im Corpus. Der Algortihmus 3.12 versucht direkt, die Sequenzen anhand des Gütekriteriums einzusetzen. Der in hier nicht aufgeführten Experimenten[3] deutlich schlechter abschneidende Algorithmus 3.13 sucht nach den Grenzen zwischen Sequenzen.

Algorithmus 3.12 (Direktes Zusammenfassen)

1. *Suche eine Sequenz, die bezüglich der Sequenzengüte am besten abschneidet und markiere sie als Sequenz.*

2. *Falls es keine solche Wortfolge gibt,* **STOP**.

[3] Das Programm *pro* realisiert diesen Algorithmus mit den Optionen *-jparse* resp. *-parse*.

3. *Rufe den Algorithmus rekursiv mit der Wortfolge links bzw. rechts der gefundenen Sequenz auf.*

Algorithmus 3.13 (Suche von Sequenzengrenzen)

1. *Falls die Länge der Sequenz klein genug ist: Fasse die Worte zu einer Sequenz zusammen,* **STOP**.

2. *Berechne für alle Zwischenräumen zwischen Worten eine Maßzahl für das Vorhandensein einer Sequenzengrenze und bestimme das Maximum dieser Maßzahl über alle Zwischenräume.*

3. *Falls dieses Maximum klein genug ist: Fasse die Worte zu einer Sequenz zusammen,* **STOP**.

4. *Zerteile die Wortfolge am Maximum und rufe den Algorithmus rekursiv auf.*

Als Maßzahl für das Vorhandensein einer Sequenzengrenze wird hier das Minimum des Maximums der Sequenzengüte der Prefixe bzw. Suffixe nach bzw. vor dem Wortzwischenraum benutzt.

Der Vorgang der Sequenzensuche kann iteriert werden, indem man beim Algorithmus 3.12 und 3.13 nur solche Sequenzen von Worten zusammenfassen darf, die keine alte, in der vorherigen Iteration gefundene Sequenz aufspaltet. Man kann dies zum einen benutzen, um eine hierarchische Struktur aufzubauen oder zum anderen, um speziell beim Algorithmus 3.13 eine Tendenz zu längeren Sequenzen zu erzwingen.

Auf den ersten Blick scheint der Algorithmus 3.13 ein globaleres Kriterium der Zusammenfassung zu bieten, und man könnte hoffen, so bessere Sequenzen zu finden. Es stellte sich aber sowohl bei der manuellen Beurteilung als auch bei den Messungen mit dem MDL Kriterium heraus, daß in der Praxis der Algorithmus 3.12 sehr viel bessere Ergebnisse liefert. Er hat weiterhin den Vorteil, daß man ihn einfacher implementieren kann, da die einzige Information, die man benötigt, um ihn durchzuführen, eine nach Sequenzengüte geordnete Liste von Sequenzen ist.

Konkret wird im folgenden daher, sofern nichts anderes gesagt wird, folgendes kombinierte Verfahren benutzt, daß als Voreinstellung im PERL -Skript *build_corpus* verwendet wird:

Algorithmus 3.14 (Verfahren zur Sequenzenbestimmung)
Zunächst wird eine Liste von n geordneten Kandidatensequenzen *bestimmt. Im Allgemeinen wird n vom Benutzer vorgegeben.*

1. *Ordne die Sequenzen, die mindestens 3 mal im Corpus vorkommen, anhand der Sequenzengüte mit dem indirekten Sequenzengütemaß mit mutual information und streiche alle Kandidatensequenzen aus der zweiten Hälfte der Tabelle.*

2. *Wende mit diesem Sequenzengütemaß den Algorithmus 3.12 mit der iterierten Variante des Sequenzensuchverfahrens und* $p = 0.5$ an.

Die iterierte Anwendung von Algorithmus 3.12 kann wie folgt vereinfacht durchgeführt werden[4]

1. *Initial besteht der Corpus aus Sequenzen, die aus jeweils einem Wort bestehen.*

[4] Diese Implementierung kann rein durch "Suchen und Ersetzen" mit regulären Ausdrücken implementiert werden und wurde effizient im PERL -Skript *chunker* umgesetzt.

2. *Führe für alle Sequenzen s enstprechend ihrer Güte geordnet aus: Fasse mit der Sequenz s alle Sequenzen im Corpus zu einer neuen Sequenz zusammen, die von s vollständig überdeckt werden .*

Das resultierenden Modell für die Sequenzensuche ist also die im Schritt 1 ersten Teil des Algorithmus 3.14 geordnete Liste von Sequenzen.

Selbstverständlich wäre es möglich, aus diesem Verfahren eine Grammatik im klassischen Sinne zu extrahieren. Der Leser kann sich aber bei einem Versuch leicht davon überzeugen, daß dies zu einer sehr länglichen Darstellung führt, die keine neuen Einsichten einbringt. Dies wird zum einen durch die den Regeln aufgeprägte Ordnung als auch durch die Möglichkeit, bereits zusammengefaßte Sequenzen erneut zusammenzufassen, bedingt. Würde man dies vernachlässigen, könnte man die einzelnen Sequenzen als Produktionen einer Grammatik betrachten, die eine Folge von Klassen auf das Symbol der Klassensequenz zusammenfaßt. Zusätzlich würden dann noch Produktionen benötigt, die Worte auf ihre Klassen reduzieren können und beliebige Sequenzen von Worten und Klassensymbolen aus dem Startsymbol erzeugen können. In dieser Grammatik wären tatsächlich auch die mit dem obigen Verfahren abgeleitete Zusammenfassungen erklärbar. Im Gegensatz zu meinem Verfahren ist die direkte Umformung in eine Grammatik aber nicht mehr deterministisch und berücksichtigt die Information über die Folgen-Teilfolgenbeziehung und die Gewichtung der Regeln nicht.

Vergleich zum Verfahren von Marcus Das Verfahren von [MM91][BMS90] ist mit dem hier vorgstellten vergleichbar. Dieser Abschnitt soll von daher aufzeigen, welches die Gemeinsamkeiten und welches die Unterschiede sind. Zunächst faßt Marcus in einem Schritt nur Sequenzen zusammen, die aus zwei Worten bestehen.

Als Zusammenhangsmaß wird im Kontext $w_1, \ldots, w_i, w_{i+1}, \ldots, w_n$ für die Sequenz (w_i, w_{i+1}) die Maßzahl

$$\sum_{k \leq i < l} c_{k,l} \mathcal{MI}(w_k \cdots w_i, w_{i+1} \ldots w_l)$$

mit nicht näher spezifizierten Gewichtungsfaktoren $c_{k,l}$ verwendet. Das Zusammenfassen der Sequenzen geschieht ähnlich wie in Algorithmus 3.12. Damit wurde erreicht, daß die Unterteilungen kontextabhängig werden, was bei dem von mir vorgestellten Verfahren auf den ersten Blick nicht zutrifft[5].

Dieser Vorteil wurde mit dem Nachteil erkauft, daß immer nur Sequenzen aus zwei Worten in einem Schritt zusammengefaßt werden können. Auf diese Weise können längere idiomatisch gebrauchte Sequenzen nur mit Mühe erfaßt werden. Es ist daher insbesondere nicht so einfach wie in dem in dieser Arbeit entwickelten Verfahren abzuschätzen, wann man Sequenzen wiederum neu klassifizieren sollte, um weiter zusammenzufassen. Ein weiterer Nachteil dieses Verfahrens ist, daß nur Maßzahlen zwischen nahe benachbarten Wortfolgen verglichen werden können und daher einige Vorkehrungen für eine korrekte Zusammenfassung der Sequenzen getroffen werden müssen, wenn man dieses Verfahren iteriert. Aus [MM91][BMS90] geht außerdem nicht exakt hervor, wie das Verfahren erweitert wird, so daß auch Folgen aus mehr als zwei Worten zusammengefaßt werden können.

[5] Da bei meinem Verfahren jede Sequenz stets mit anderen Sequenzen, die ein Stück verschoben sind, in Konkurrenz steht, wird auf diese Weise der Kontext auch in meinem Verfahren berücksichtigt. Wenn man z.B. die Sequenz `A B C` durch `B C` zusammenfassen möchte, die Sequenz `A B` aber eine bessere Güte hat und vorher zusammengefaßt wird, kann es nicht mehr zur Zusammenfassung von `B C` kommen. Damit hatte der Rechtskontext `A` der Sequenz `B C` die Zusammenfassung verhindert.

```
+um+_+ +um+ +ls+
       +um+ +h#+
       +um+ +um+
+um+_+ +um+ +paper+
       +um+ well
       we i
       probab think
we we
need need
for_pr for to
       probab meet
for_th for for
       this another
+um+ +um+
noon_n noon two
        noon hours
        +um+ +um+
for_ar for to
       arrang discuss
this_a this this
       appoin matter
furthe further
+um+_+ +um+ +um+
       +um+ +ls+
       +um+ +h#+
what's what's
this_a this a
       additi convenient
time time
for_pr for for
       probab like
this_n this a
       noon two
       noon hour
slot slot
within within
februa februa the
        februa next
noon_n noon couple
        noon weeks
for_pr for for
       probab you
       +um+ $1
       +um+ +paper+
```

```
+nonhu +nonhu +ls+
       +nonhu +h#+
+nonhu +nonhu +um+
       +nonhu +paper+
       hi well
i i
think think
we we
need need
to to
meet meet
for for
anothe another
+um+ +um+
two two
hours hours
+um+ +um+
to to
discus discuss
this this
matter matter
furthe further
+nonhu +nonhu +um+
       +nonhu +ls+
+h#+ +h#+
what's what's
a a
conven convenient
time time
for for
like like
a a
two two
hour hour
slot slot
within within
the the
next next
couple couple
weeks weeks
for for
you you
+nonhu +nonhu $1
       +nonhu +paper+
```

Abbildung 3.8: **Vergleich der Sequenzen unter verschiedenen Gütekriterien:** Diese Abbildung zeigt das Ergebnis von Zusammenfassungen von einem Satz nach dem Standardverfahren bei 100 Sequenzen und 25 Klassen. Die Worte aus dem Originalcorpus sind **fett** gedruckt. Durch eine Einrückung nach rechts wird eine Zusammenfassung von Worten (bzw. Sequenzen)zu einer Sequenz dargestellt. Die Sequenz wird jeweils durch ein neues Symbol dargestellt, daß aufgrund der Länge der entstehenden Symbole nur mit einem Prefix von 6 Zeichen dargestellt werden kann. Als Sequenzengütemaße wurden links die *mutual information* mit der iterierten Markierungsfrequenz und rechts das Maß von Suhotin direkt genommen. Es fällt zum einen auf, daß rechts trotz gleicher Sequenzenzahl weniger Sequenzen im Text gefunden wurden. Dies bedeutet aber auch, daß beim schlechteren Maß auch schlechtere Sequenzen verwendet werden und damit die Genauigkeit der Zusammenfassung leidet. Zum anderen sind im rechten Teil nur uninteressante Zusammenfassungen gefunden worden, während im linken Teil auch syntaktisch motivierte Zusammenfassungen zu sehen sind (z.B. **+paper+ well i think**).

Die Abbildung kann nach einer erfolgten Corpusanalyse mit dem PERL -Skript *build_corpus* aus einer *.parsed* Datei mit dem PERL -Skript *pretty_parse* erzeugt werden. Das Skript bietet mit der Option *-tab* die Möglichkeit, direkt die obige Abbildung auszugeben.

3.3 Kombination von Distributions- und Sequenzenanalyse

Ein wesentlicher und originärer Schritt in dieser Arbeit stellt die Kombination der Operatoren Klasseneinteilung und Verkettung dar. Die Sequenzensuche hat bislang nur auf Wortebene operieren können und man muß annehmen, daß die Zusammenfassung sehr langer Sequenzen so nur schlecht funktionieren kann. Die Wortklassifikation könnte durch die Einbeziehung von Information aus weitreichenden Verbindungen verbessert werden, die durch die langen Sequenzen in der Nähe des Wortes gegeben sind.

Für diese Arbeit ist insbesondere die genauere Modellierung der Zusammenfassung langer Sequenzen interessant. Die Wortklassifikation bedarf, wie man in den vorigen Kapiteln deutlich sieht, vorerst keiner weiteren Verbesserung, obwohl dies ein für die Zukunft lohnendes Projekt sein könnte.

3.3.1 Verbesserung der Sequenzensuche durch Distributionsanalyse

Die Verbesserung der Sequenzensuche durch ein iteriertes Verfahren sieht im Groben so aus

1. Berechne ein Klasseneinteilung der Worte (bzw. bereits zusammengefaßter Wortsequenzen) im Corpus.
2. Suche die Sequenzen auf der Basis dieser Klasseneinteilung und fasse nur wenige, aber sehr sichere Sequenzen zusammen.

Eine bottom-up Zerteilung einer Sequenz kann ebenfalls als ein zweistufiges System interpretiert werden, das iteriert wird:

1. Suche nach einer Sequenz, die als rechte Seite einer Grammatikregel gegeben ist (Sequenzensuche).
2. Ersetze dies Sequenz durch einen Namen, der als linke Seite einer Grammatikregel gegeben ist (Klasseneinteilung).

Die Schritte der Klassifikation und der Sequenzensuche kann man wiederholen und gelangt so zu einem iterativen Verfahren. Genauer hat man das im PERL -Skript *build_corpus* implementierte Verfahren:

Algorithmus 3.15 (Iterierte Sequenzensuche)

1. *Klassifiziere die Einheiten des vorliegenden Corpus mit Algorithmus 3.6*
2. *Suche die Sequenzen auf dem Corpus, bei dem die Worte durch ihre Klasse ersetzt wurde mit dem Algorithmus 3.14.*
3. *Bilde einen neuen Corpus, in dem alle in Sequenzen vorkommenden Worte durch ihre jeweiligen Klassen ersetzt werden.* Optional: *Ersetze auch alle Worte, die nicht in den Sequenzen vorkommen, durch ihre Klassen*
4. *Gehe zum Schritt 1 oder breche ab*

Der Algorithmus wird in aller Regel für eine feste Zahl von Iterationen durchgeführt. Falls aufgrund der starken Zusammenfassung des Corpus keine oder zu wenige Sequenzen gefunden werden, kann ebenfalls abgebrochen werden.

specif specif no_i'm +ls+_+ you_li +h#+ +h#+
+uh+ +uh+
okay **okay**
i_gues i **i**
guess **guess**
seem i **i**
need **need**
to **to**
meet **meet**
with **with**
you **you**
+ls+_m +ls+_m for **for**
aftern about **about**
two **two**
hours **hours**
though during **during**
the **the**
week **week**
it_is +um+_m or_wel here's +um+ **+um+**
+muell **+muell+**

Abbildung 3.9: **Strukturfinden in einsprachigen Corpora:** Der Vorgang der Klasseinteilung und Sequenzensuche wurde acht mal wiederholt und für zwei unterschiedlich Sequenzenzahlen durchgeführt. Die Anzahl der verwendeten Klassen beträgt 400, die Anzahl der Sequenzen 4000. Die Sequenzenbestimmung wurde mit dem Verfahren 3.14 durchgeführt, als Kontext wurde $[-1,-1][1,1]$ gewählt. Die Darstellungsmethode ist dieselbe wie in Abbildung 3.8.

Im vorletzten Schritt werden die in den Sequenzen vorkommenden Worte auf jeden Fall durch ihr Klassensymbol ersetzt. Dies ist die minimal erforderliche Ersetzung, da sonst die Anzahl der Symbole sehr stark anwachsen würde. Die optionale Ersetzung aller Worte durch ihr Klassensymbol durchzuführen hat sich in den Experimenten als nicht vorteilhaft erwiesen.

3.3.2 Messung der Güte des Modelles

Im vorangegangen Kapitel wurde eine Methode zur Korpusanalyse vorgestellt, deren Ergebnisse gut durch menschliche Beurteilung der Ergebnisse zu rechtfertigen sind. Dies ist für die Abfassung einer wissenschaftlichen Arbeit nicht allein ausreichend, da

- die zu beurteilenden Datenmengen groß sind und unvollständig bleibt
- die Zeit zur Beurteilung eines Modelles lang ist
- die Bewertung subjektiv ist
- der Vergleich mit andereren Methoden nicht möglich ist
- die Ergebnisse nur schwer kommunizierbar sind

Aus diesen Gründen ist die Bereitstellung eines objektivierbaren Maßes wünschenswert. In wieweit dieses Maß dann im Endeffekt wirklich die Analysegüte adäquat beschreibt, kann auf verschiedene Weisen geklärt werden:

- decken sich die Meßdaten mit den menschlichen Einschätzungen
- lassen sich die Meßdaten verschiedener Modelle vergleichen
- deckt sich die vorhergesagte Analysegüte mit Maßen, die in späteren Verarbeitungsschritten verwendet werden

Mit der Solomonoff Komplexität [Sol64a][Sol64b] ist ein Werkzeug gegeben, daß solche Aussagen für induktive Inferenzsysteme ermöglicht. Es basiert auf einer wissenschaftstheoretischen Einsicht, die auch als *Occams Razor* bekannt ist: Wenn Du mehrere Erklärungsmodelle für Deine Beobachtungen hat, dann wähle das einfachste.

Das Minimum Description Length Principle (MDLP) In der Arbeit [Ris89] wurde das Minimum Description Length Prinzip, in dieser Arbeit stets kurz mit MDLP bezeichnet, eingeführt, das sich als grobe Approximation der Solomonoff Komplexität betrachten läßt. Dieses Maß läßt sich sowohl dazu verwenden, die Güte einer Analyse zu beschreiben als auch die Modellauswahl direkt durch dieses Maß zu steuern [Ell91][CH94]. Ansätze, die versuchen, dieses Kriterium direkt auszunutzen, zeichnen sich vor allem durch einen hohen Rechenbedarf aus. Die durch [Ell91] beschriebenen Phonemklassifikationen lassen sich mit den in dieser Arbeit beschriebenen Verfahren sehr leicht bestimmen und beurteilen. Das Verfahren [CH94], das Teilgraphen nach dem MDLP sucht und zusammenfaßt, wurde testweise auch auf Parsebäumen von Teilkorpora eingesetzt mit dem Ziel, aus bereits identifizierten Bäumen interessante Strukturen abzuleiten. Es konnte aber nicht genutzt werden, da die Teilkorpora, die noch in ein paar CPU-Stunden oder Tagen analysierbar waren, viel zu klein waren. In dieser Arbeit wird nur das Ergebnis einer Theoriebildung auf der Basis seiner Description Length (DL) beurteilt.

Das MDLP besagt, daß die Theorie T^* die beste ist, die zusammen mit den unter der Annahme dieser Theorie beobachteten Corpus C die geringste Entropie (auch Descrption Length (DL) genannt) besitzen, oder aber

$$T^*(C) = argmin_T \mathcal{H}(T) + \mathcal{H}(C|T) = argmin_T DL(C,T)$$

Modell des Corpus Zur genaueren Spezifikation der DL muß noch aufgeschrieben werden, wie die Theorie und der Corpus unter der Voraussetzung der Theorie übertragen[6] werden sollen. Das Modell des Corpus wird folgendermaßen erzeugt

1. Berechne mit der Distributionsanalyse eine Klasseneinteilung der Worte.
2. Suche mit der Sequenzenanalyse Sequenzen von Wortklassen und markiere diese im Originaltext.

Die Übertragung des Modelles geschieht folgendermaßen

1. Übertrage die Sequenz der Wortklassen aller Worte des Textes, ersetze aber jede gefundene Sequenz von Wortklassen durch ein Symbol.
2. Übertrage für jede Klasse jeweils den Subcorpus, der durch das Weglassen aller Worte entsteht, die nicht zur jeweiligen Klasse gehören.
3. Übertrage eine Zuordnung von Worten zu Klassen, indem zu jedem Wort die zugehörige Klasse übertragen wird.

[6] Der Begriff der Übertragung wird im Zusammenhang mit der DL für den Vorgang der Kodierung in einem in gewisser Weise optimalen Code verwendet

4. Übertrage eine Liste mit den Sequenzen, also eine Liste von Listen von Wortklassen.

[Ell91] benutzt ein sehr ähnliches Verfahren und nennt die Subcorpora im zweiten Schritt auch *planes*. Die Idee bei der Übertragung der *planes* ist es, daß in diesem Subcorpus nur Elemente einer Klasse übertragen werden und die bei der Kodierung ausgenutzt wird. Im Unterschied zu [Ell91] können bei diesem Modell aber Sequenzen an einer beliebigen Stelle und beliebig oft in einem Textstück auftreten. Für jede Sequenz wird zunächst die Länge n der Sequenz übertragen ($\log_2(n)bit$) und anschließend sequentiell für jedes Symbol des Corpus die entsprechende Kodierung. Für die Kodierung eines Symboles in einem Subkorpus mit l Zeichen kann man zwei Möglichkeiten wählen

- nehme alle Symbole als gleichwahrscheinlich an und übertrage jedes Symbol w_i mit der entsprechenden Kodelänge ($\log_2(l)bits$)
- übetrage jedes Symbol w_i mit seiner optimalen Kodelänge ($\log_2(p_{w_i})bit$)

Gegen die zweite Kodierungsvariante könnte man ins Feld führen, daß hiermit die Eigenheiten des vorhandenen Korpus zu stark kodiert werden, da der Kode bereits unter der Annahme der Wortwahrscheinlichkeiten des Korpus gemacht wurde. Dieses Argument zielt vor allem deshalb ins Leere, da der Untersuchungsgegenstand nicht wie bei der Perplexität die Vorhersage von Text ist sondern vielmehr die Güte des Modelles. Wenn man aber ausschließlich die erste Kodierungsalternative zulassen würde, würden Verkettungen von Worten stark überbewertet, die selten im Korpus vorkommen.

Bei der iterierten Sequenzensuche wird zur Übertragung des Corpus aus dem Schritt 1. ein neues Corpusmodell verwendet, das entsprechend der obigen Beschreibung konstruiert wird. Die hier gegebene Formulierung setzt dabei voraus, daß alle Worte durch ihr Klassensymbol ersetzt wurden (Anwendung des *optionalen* Teiles in Algorithmus 3.14). Es ist genauso denkbar, die Worte, die nicht in Sequenzen vorkommen, nicht durch ihr Klassensymbol zu ersetzen. Das erste Verfahren sei als das *reine* Verfahren bezeichnet, das zweite als das *direkte* Verfahren. In dieser Arbeit wurde immer das *reine* Verfahren benutzt, sofern nichts anderes erwähnt wird. Das Skript *chunker* kann mit der Option *-mdl* eine solche Analyse für eine Iteration durchführen, das Skript *build_corpus* benutzt das Skript *mdl_extract*, um das Corpusmodell für die iterierte Sequenzensuche zu extrahieren. Das *direkte* Verfahren kann durch die Option *allsmall* beim Skript *chunker* benutzt werden.

Man kann das Verfahren von oben bei der Anwendung der zweiten Kodierungsvariante und unter der Voraussetzung, daß Sender und Empfänger jeweils eine vollständige Liste haben, die alle Worte des Corpus enthalten, entscheidend vereinfachen. Diese Annahme ist realistisch, da sie in der ersten Iteration der Annahme entspricht, daß nur bekannte Worte und das unbekannte Wort übertragen werden und bei einem iterierten Verfahren dann automatisch erfüllt ist.

1. Übertrage einen Corpus aus Wortklassensymbolen und Sequenzensymbolen, der sich in folgende Teilcorpora untergliedert
   - Übertragung der Wortklassen aller Worte aus der Liste der Worte
   - Übertrage für alle Sequenzen jeweils der Reihe nach die Wortklassen, die in ihnen vorkommen
   - Übertragung die Klassen der Worte bzw. der gefundenen Sequenzensymbole
2. Übertrage eine Liste von Zahlen, um die obigen Teilsequenzen trennen zu können

Anzahl der Klassen	100 Sequenzen	1000 Sequenzen	4000 Sequenzen
25	185		
50	151		
100		102	176
200		91	132
400		90	132
600		91	131

Abbildung 3.10: **Description Length des englischen Corpus abhängig von den Modellparametern:** Die Description Length (angegeben als $2^{DL(C)}$) auf dem englischen Corpus) ist ein wesentlich härteres Maß als die unten folgenden Perplexität und bestraft vor allem die Verwendung von vielen Sequenzen. Nach der ersten Iteration ist keine weitere Verbesserung der DL durch die Anwendung der Corpusanalyse mehr möglich, obige Werte sind außerdem wesntlich schlechter als die entsprechenden Perplexitätsangaben auf dem gleichen Corpus. Nach dieser Tabelle müßte man den englischen Corpus mit 1000 Sequenzen und 400 Wortklassen analysieren.

3. Übertrage die Worte unter der Voraussetzung ihrer Wortklasse.

3.4 Anwendungen der unüberwachten Analyse

Die vorgestellten unüberwachten Methoden zur Corpusanalyse stellen kein rein akademisches Produkt für sich dar. Tatsächlich stellt die schnelle Erlernbarkeit von syntaktischer Struktur aus unstrukturierten Text ohne manuelle Intervention ein für sich interessantes Resultat dar. In diesem Abschnitt sollen einige konkrete Anwendungen im Bereich der Spracherkennung und -analyse besprochen werden, die im Janussystem eingesetzt werden können oder bereits getestet wurden.

3.4.1 Sprachmodellierung in der Spracherkennung

Eine gute Sprachmodellierung stellt ein entscheidendes Werkzeug für den Erfolg eines Spracherkenners dar. Die in der Spracherkennung am häufigsten verwendeten Modelle sind dabei die sogenannten *n-gramme*, Sprachmodelle die versuchen, die Wahrscheinlichkeit des nächsten Wortes auf der Basis des Wissens über die letzten $n-1$ Worte vorherzusagen. Im allgemeinen wird in der Sprachmodellierung die Leistung eines Modelles im Maß der Perplexität PP einer diskreten Zufallsvariablen X bezüglich einer Wahrscheinlichkeitsdichte p (geschätzt auf der Testmenge) bezüglich der Wahrscheinlichkeitsdichte q (geschätzt auf der Trainingsmenge) gemessen.

$$PP(p,q) = 2^{LP(p,q)} = 2^{\sum_{x \in X} p(x) \log q(x)} = 2^{\mathcal{H}_p(X) + d(p\|q)}$$

Wenn man das oben beschriebene Modell des Corpus verwendet, um die Daten auf Testdatenmenge zu übertragen und dabei zwar nicht das Modell mitüberträgt sondern statt dessen die Länge der Kodierungen auf der Basis der Wahrscheinlichkeiten der Trainingsdatenmenge erfolgt, kann man die oben erzeugten Modelle unmittelbar auf der Basis der Perplexität mit den n-gramm Modellen vergleichen. Dieser Vergleich ergibt insbesondere für den englischen und den deutschen Terminabsprachecorpus (ESST und GSST), daß die in dieser Arbeit erzeugten Modelle ein sehr viel besseres Modell dieser Corpora bilden.

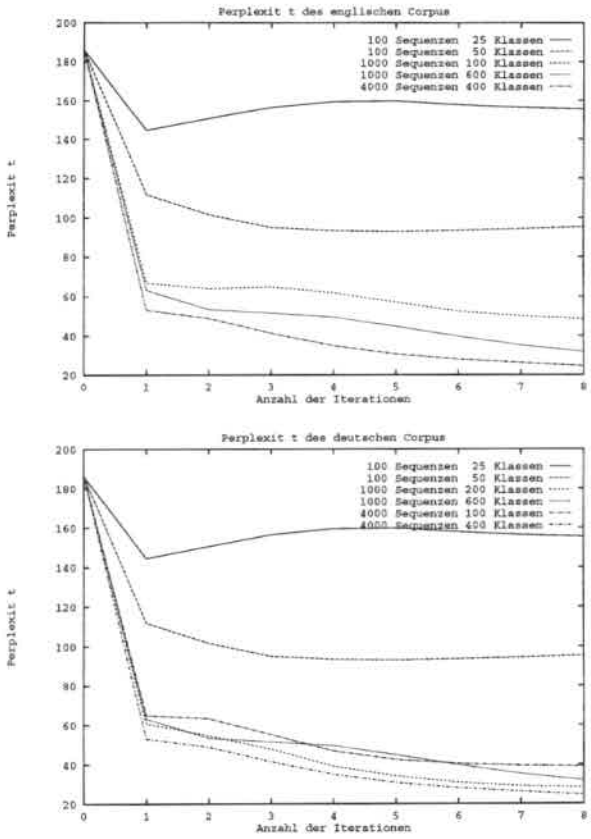


Abbildung 3.11: **Perplexität der Corpora im Verlauf der Iterationen:** Die Abbildungen zeigen die Perplexität der Corpora in Abhängigkiet von der Anzahl der Iterationen. Das beste Modell ist sowohl bei englischen wie beim deutschen Corpus das Modell mit 4000 Sequenzen und 400 Wortklassen. Sehr deutlich ist, daß die größte Reduktion im ersten Schritt erfolgt und danch nur noch leichte aber stetige Verbesserungen zu beobachten sind. Für den ensglischen Corpus ergibt sich mit diesem Modell nach 8 Iterationen eine Perplexität von 24.6, für den deutschen eine Perplexität von 40.9. Damit sind diese Modelle deutlich besser als die auf den gleichen Corpora trainierten Bigramme und Trigramme, auch wenn diese klassenbasiert bestimmt wurden.

Anwendung bei der Dialogmodellierung Bei der Domäne der Terminabsprachedialogen kann man verschiedene Phasen des Dialoges unterscheiden wie z.B. die Begrüßung, den Vorschlag eines Termins, Beschränkung auf einen Zeitraum, die Annahme/Zurückweisung eines Terminvorschlages und die Verabschiedung. Während dieser verschiedenen Phasen des Dialoges kann man unterschiedliche, spezialisierte Sprachmodelle wählen, die die jeweilige Phase präziser modellieren können.

In der Arbeit [WW94] werden Markov Ketten und Hidden Markov Modelle [Rab89] verwendet, um Dialogzustände zu bestimmen. Da die Anzahl der Worte im Corpus und die Anzahl der zu schätzenden Dialogzustände im Verhältnis zur Größe des Corpus sehr groß waren, war die Schätzungen der Modellparameter nur noch mit große Unsicherheit möglich. Durch die Verwendung von Wortklassen statt der Worte konnten zum einen die Anzahl der Modellparameter deutlich reduziert werden und zum anderen besteht die Hoffnung, daß duch die Klassifikation die Generalisierungsfähigkeit des Modelles verbessert wird.

Tatsächlich sind die Resultate durch die Verwendung von Wortklassen erheblich verbessert worden und in [WW94] wurde darüberhinaus festgestellt, daß sich die Verwendung von handerstellten Klassen im Vergleich zu den hier vorgestellten automatisch generierten kaum rentiert. [WW94] bemerkt darüberhinaus, daß das Hidden Markov Modell in gewisser Weise bestimmte im Corpus häufig vorkommenden Sequenze lernt.

Die Einführung von Sequenzen in ein Markov Modell wie in [WW94] hat keine Erfolg gebracht, ebenso die Nutzung von Neuronalen Netzen, die in der Regel 5-8% schlechter abschnitten als das Markov Modell. Es wurden dazu TDNN's [WHH+89], Gamma Netze [dVP92] und RAAM verwendet. Das einzige erfolgversprechende Experiment mit neuronalen Netzen wurde mit von Hand in Sprechakte unterteilten Äußerungen gemacht. Dabei wurde die Sequenz als ganzes in das Neuronale Netz eingegeben. Bei Verwendung von Wortklassen hat sich eine Performanz ergeben, die ca. 3-5%-Punkte über dem Ansatz mit dem Markov Modell liegt. Die Verwendung der Sequenzen brachte hier eine geringfügige Verbesserung. Es bietet sich damit an, einen neuronalen Parser mit einer ähnlichen Eingaberepräsentation zu testen. Aufgrund des sehr dünnen Datenmaterials, das mit 12 Dialogen gegeben ist, möchte der Autor hier allerdings jeden Vergleich vermeiden. *Das einzig konsistente Ergebnis über alle Experimente war, daß das RAAM das beste aller getester Netze ist und die Eingabe eines Sprechaktes als ganzes in das Netz die besten Resultate liefert.*

Insbesondere die Markierungen der Sprechakte, die per Hand erfolgte, wiederspricht oft den Intentionen des Autors. Es schließt sich [WW94] an, die im zweiten Teil die Sprechakte durch ein Hidden Markov Modell automatisch bestimmen läßt. Dieses Verfahren könnte man ebenfalls mit den hier vorgestellten Methoden machen – dazu müßte man die bisher durch die Analyse nicht übergehbaren Dialoggrenzen aufheben und auch Äußerungen klassifizieren und zusammenfassen dürfen.

Kapitel 4

Übersetzung mit statistischen Methoden

Unter dem Oberbegriff *Übersetzung mit statistischen Methoden* finden sich in der Literatur unterschiedliche Verfahren, die bereits in der Einführung auf Seite 10 kurz vorgestellt wurden. Bereits in der Einführung wurde erwähnt, daß keines dieser Verfahren eine vollständige und realistische Implementierung eines statistischen Übersetzungssystemes darstellt, das aus einem zweisprachigen Parallelcorpus[1] ohne oder mit geringer menschlicher Intervention ein automatisches Übersetzungssystem erstellen könnte. Dieses Kapitel soll die Versuche beschreiben, wie man ein solches System erstellen kann.

Prinzipiell muß ein solches Übersetzungssystem aus zwei Stufen bestehen, der Generierung der Übersetzungsbeispiele und dem Lernen der Übersetzungskomponente. Die Notwendigkeit für eine Trennung der Beispielerzeugung und des Lernens der Übersetzungskomponente wird bei der detailierteren Darstellung und Kritik des Wortalignment[2] nach [BPPM93a] deutlich werden: Die Methoden zur Generierung von Beispielen werden in der Regel sehr viel einfacherere Methoden benutzen als diejenigen, die aus den Beispielen den Transfer herstellen. Ein Modell wie das von [BPPM93a] zum Wortalignment benötigt nicht zwingend die Fähigkeit, die korrekten Übersetzungsvarianten eines Wortes aus dem Kontext zu bestimmen, während die zugehörige Methode zum Transfer dies unbedingt können muß [BPPM91]; ob aber die derzeit üblichen Vereinfachungen beim Alignment tatsächlich sinnvoll sind, kann bestritten werden (siehe Kapitel 4.1). Die Grundidee dieser Arbeit bestand nun darin, die Methoden des Example Based (EBMT) und der Transfer Driven (TDMT) Übersetzung [FI92] [SI92] [Iid93] [Bri93] [3], die keine Methoden zur automatischen Beispielgenerierung beinhalten, mit Methoden des Wortaligment zur Beispielgenerierung zu verkoppeln.

Dieser Plan ist vor allem daran gescheitert, daß die Ergebnisse im Bereich des Alignment keine Hoffnung zuließen, eine brauchbare Übersetzungskomponente zu trainieren. In dem Kapitel über Alignment sollen daher die auftretenden Probleme beleuchtet werden und berichtet von den getesteten Methoden. Das Kapitel über das Lernen der Übersetzungskomponente stellt eine reine Literaturarbeit dar. Eine

[1] Unter einem Parallelcorpus soll verstanden werden, daß ein Text in mehreren Sprachen vorliegt. Für diese Arbeit stand ein spontansprachlicher Corpus mit englischen Terminabsprachedialogen und dessen deutscher Übersetzung zur Verfügung.

[2] Der englische Begriff *Alignment* bezeichnet den Vorgang, zwei Texte, Wortfolgen etc. in "Übereinstimmung" zu bringen bzw. zuzuordnen und wird direkt übernommen, da dem Autor keine gute Übersetzung bekannt ist.

[3] Die Arbeit [Bri93] beinhaltet noch nicht die Anwendung seiner Verfahren auf die Übersetzung, die aber im Kapitel 4.2 deutlich wird.

gesondertes Kapitel beschäftigt sich mit Methoden zur Übersetzung, die ohne ein Alignment auskommen.

4.1 Erzeugung von Übersetzungsbeispielen durch Alignment

Für diese Arbeit war ein englischer, spontansprachlicher Corpus aus Terminabsprachedialogen mit handerstellten Übersetzungen gegeben. Durch die existierende Zuordnung der einzelnen Äußerungen zu ihren Übersetzungen mußte diese Zuordnung nicht automatisch erstellt werden. Die Methoden zur Zuordnung von Sätzen, auch Satzalignment genannt [KR93] [CDG+93b] [GC93] [CDG93a] [BLM91] mußten also nicht zur Anwendung kommen. Die zu lösende Aufgabe war es also, die Worte bzw. Satzteile aus einer englischen Äußerung den zugehörigen Worten bzw. Satzteilen der deutschen Übersetzung zuzuordnen und soll als Wortalignment bezeichnet werden.

Das Modell von [CDG93a] kann hier nicht zur Anwendung kommen, da es in erster Linier zum Auffinden einzelner, seltener Worte und deren Übersetzungen gedacht ist[4] – dieses Problem ist deutlich leichter zu lösen, da die Übersetzungen von seltenen Worten ebenfalls selten vorkommen und daher eine Zuordnung relativ leicht ist.

Die Modell [BPPM93a] zeichnet sich durch eine Überbetonung einer Wort-für-Wort Übersetzung aus, die bei spontan gesprochener Sprache problematisch ist, wie man an den vergleichsweise einfachen Beispielen in der Abbildung 4.1 sehen kann. [BPPM93a] formuliert dazu eine Folge von Modellen, die zunehmend komplizierteren Übersetzungsmodellen entsprechen. Die Modellparameter werden durch einen EM-Algorithmus (siehe Kapitel 2.5) geschätzt. Dabei sind nur die einfachsten Modelle (Modell 1 und 2) ohne großen Aufwand zu berechnen. Die komplizierteren Modell (Modell 3 bis 5) sind zunehmend schwieriger zu berechnen und man darf bezweifeln, daß Modelle, die komplexer sind als das Modell 5, dem EM-Algorithmus noch praktisch zugänglich sind. Selbst die "kompliziertesten" Modelle 4 und 5 sind aber sehr einfache Modelle von Übersetzung und besitzen lediglich folgende Parameter

Übersetzungswahrscheinlichkeiten Für jedes Wort wird die Wahrscheinlichkeit bestimmt, Übersetzung eines anderen zu sein.

Generierungswahrscheinlichkeiten Für jedes Wort wird die Wahrscheinlichkeit bestimmt, daß es in $1, 2, \ldots$ Worte der anderen Sprache übersetzt wird. Es wird eine Sonderbehandlung für die Nichtübersetzung eines Wortes eingeführt.

Verschiebungswahrscheinlichkeiten Für jedes Wort bzw. Wortgruppe wird angegeben, wie wahrscheinlich sie um eine bestimmte Anzahl von Worten bei der Übersetzung verschoben werden.

Die Selektion der Wortgruppen erfolgt bei [BPPM93a] auf der Basis einer Analyse des einsprachigen Corpus. Dazu wurden die Wortklassifikationsmethode [BdP+92] benutzt, die in der Beobachtung 3.8 mit den im Rahmen dieser Arbeit entwickelten Methoden im Beziehung gebracht wurden. Welche Methoden allerdings zur Identifikation der Sequenzen benutzt wurden, kann aus [BPPM93a] nicht abgelesen werden.

Das Modell von Brown kann in dieser Arbeit auf keine Fall direkt eingesetzt werden, da folgende Probleme auftreten

[4] Die Anwendung dieses Programmes ist die Generierung von Übersetzungsbeispielen für technische Ausdrücke in Computerbeschreibnungen, die im Vergleich zu anderen Worten sehr selten vorkommen und soll menschliche Übersetzer unterstützen.

Texttyp	Beispiel	Bemerkung
Original	I'm writing in my calendar at this very moment two to four, let's meet in your office.	Satz ist relativ kurz, kaum Umstellung der Konstituenten zur deutschen Übersetzung, keine Fehler des Sprechers.
Interlinear-übersetzung	Ich schreibe in meinen Kalender, gerade jetzt, zwei bis vier. Treffen wir uns in Ihrem Büro.	Fast wörtliche Übersetzung, nur begrenzt akzeptabel.
Mündliches Deutsch	Ich schreib mir es gerade im Kalender auf, zwei bis vier, bei Ihnen im Büro?	Sehr gute Übertragung, Zuordnung zum Original auf Wortebene schwierig.
Original in Schriftsprache	I am just writing down a date from two to four in my calender. I suggest, that we meet in your office.	Umstellung der Satzstruktur, Einfügung neuer Teile.
Deutsche Schriftsprache	Ich trage jetzt einen Termin von zwei bis um vier Uhr in meinen Kalender ein. Ich schlage vor, daß wir uns in Ihrem Büro treffen.	Ähnlichkeit zum Original in Schriftsprache hoch.
Zwischenform	Ich bin gerade dabei in meinen Kalender von zwei bis um vier einzutragen, treffen wir uns doch in Ihrem Büro.	Für das Projekt verwendeter Übersetzungsstil.

Abbildung 4.1: **Übersetzungsstile für spontane Sprache:** Zu Beginn einer Übersetzungsarbeit muß festgelegt werden, welcher Übersetzungsstil vom (menschlichen) Übersetzer gewählt werden soll. Dieser Stil muß so gewählt werden, daß die entsprechende Strategie auch vom Übersetzungssystem erfolgreich gewählt werden kann und die Zuordnung der einzelnen Übersetzungsteile leicht möglich ist. Das obige Beispiel ist ein vergleichsweise einfaches, da der Satz nicht sehr lang ist, keine weiten Umstellungen in der deutschen Übersetzung nötig sind und keine Fehler des Sprechers vorliegen. Der für dieses Projekt gewählte Übersetzungsstil stellt eine Zwischenform dar und versucht, die Konstituentenstruktur weitgehend zu erhalten, sofern dies möglich ist, ohne ein korrektes gesprochenes Deutsch zu verlassen. Es besteht dadurch insbesondere die Hoffnung, sententielle Syntagmen in einem Template-basierten Ansatz zu erhalten (siehe Abschnitt 1.2).

kleiner Corpus [BPPM93a] arbeitet auf den gut formulierten, umfangreichen Französisch / Englischen Parlamentsprotokollen (Hansards) und beschränkt sich auf Sätze mit weniger als 30 Worten. Selbst unter diesen Restriktionen ergeben sich ca. 1.7 Millionen Trainingssätze, um die große Parameterzahl im Modell zu schätzen. Für das Training stehen hier dagegen ca. 1000 Äußerungen zur Verfügung, von denen nur etwa die Hälfte weniger als 30 Worte enthielten.

implizite Sprachabhängigkeiten Auch wenn [BPPM93a, Seite 296] dies ironisch bestreitet enthält sein Modell implizite Annahmen über das verwendete Sprachpaar. Ein Englisch - Deutsches Übersetzungssystem würde mit Sicherheit stark davon profitieren, speziell auf die Verbendstellung des Deutschen und die Abspaltung von Prefixen bei Verben orientierte Parameter einzuführen.

Let's see Schaun wir mal	next tuesday nächsten Dienstag	good ginge es gut	it world have to es müßte dann	be	allerdings
in the afternoon nachmittags	sein	or or oder	aber	thursday thursdays Donnerstag Donnerstag	good geht gut

Abbildung 4.2: **Alignment bei spontaner Sprache:** Die Abbildung zeigt einen einfachen Satz, bei dem die Zuordnung der Übersetzung zum Original ohne große Umstellung der Konstituenten möglich ist. Trotz dieser einfachen Struktur kann man sehen, daß eine Wort zu Wort zuordnung fast unmöglich ist, vielmehr kann man nur Satzstücke zuordnen. Größere Schwierigkeiten treten dann auf, wenn die Konstituenten über große Entfernungen verschoben werden.

spontan gesprochene Sprache Wie in Abbildung 4.2 vorgeführt wird, kann bei spontan gesprochener Sprache oft keine reine Wort zur Wort Zuordnung angegeben werden, einzelne Phrasen müssen als feststehende Redewendungen betrachtet werden und auch als solche übetragen werden. Zusätzlich ergibt sich das Problem, daß bei spontaner Sprache oft tiefe, weitreichende Einbettungen gemacht werden, die bei der Übersetzung oft wieder aufgelöst werden oder zu weiten Verschiebungen einer ganzen Sequenz führen.

Als mögliche Verbesserungen und Anpassung auf das gegebene Übersetzungsproblem sind im Laufe der Arbeit mehrere Ideen verfolgt worden.
Zum einen wurde [BPPM93a, Kapitel 6.2] gefolgt, daß in zukünftigen, komplexeren Übersetzungsmodellen in jeder Interation des Alignmentverfahrens nur noch die jeweils wahrscheinlichste Wortzuordnung innerhalb eines Satzes, das Viterbi-Alignment, betrachtet werden soll. Zusätzlich zu bisherigen Verfahren sollen die Übersetzungsparameter für einzelne Worte durch die Bildung von zweisprachigen Wortfeldern verbessert werden und die Übertragung von Phrasen durch die einsprachige Corpusanalyse und ein spezielles Übersetzungsmodell unterstützt werden. Die verwendeten Methoden werden in den nächsten Abschnitten vorgestellt.

Einfaches Modell zur Erstellung von Wörterbüchern Im Gegensatz zu [BPPM93a] sollen keine Übersetzungswahrscheinlichkeiten bestimmt werden, sondern lediglich, welches Wort aus der einen Sprache in welches Wort der anderen Sprache übersetzt wird und es soll eine Güte dieser Übersetzungsmöglichkeiten angegeben werden. Dieses Modell ist deutlich einfacher zu bestimmen, da man nicht zwingend ein wahrscheinlichkeitstheoretisches Maß benutzen muß – eine Übersetzungswahrscheinlichkeit stellt andersherum ein Gütemaß dar. Das Verfahren kann man in drei Phasen aufteilen:

1. Zähle, wie oft ein Wort w_2 aus der zweiten Sprache an Stellen vorkommt, an denen es eine Übersetzung des Wortes w_1 aus der ersten Sprache ist bzw. nicht ist (w_1 und w_2 in Übersetzungs bzw. Nicht-Übersetzunspositionen stehen).

2. Bestimme aus diesen Werten mit einem Zusammenhangsmaß für binäre Ereignisse eine Übersetzungsgüte für alle Wortpaare (siehe Abschnitt 2.6) .

3. Ordne die Liste der Wortpaare nach der Übersetzungsgüte.

Die erste Frage, die man sich stellen kann, ist, wann zwei Worte in Übersetzungspositionen stehen, obwohl man noch kein Wortalignment bestimmt hat. In dieser Arbeit war immer klar, welche Äußerungen zusammengehören, damit ist das einfachste denkbare Maß, daß zwei Worte genau dann in einer Übersetzungsposition stehen, wenn diese in einander zugeordneten Äußerungen vorkommen. Man kann

dieses Modell etwas verbessern, indem man davon ausgeht, daß in langen Äußerungen der Beginn (das Ende) der Äußerung in der einen Sprache tendentiell eher in den Beginn (das Ende) der zweiten Äußerung übersetzt wird – man unterstellt also ein tendentiell lineares Wortaligment. Um dies zu implementieren, wurde eine Wahrscheinlichkeit eingeführt, die angibt, ob sich zwei Worte in einer Übersetzungsposition befinden. Diese Wahrscheinlichkeit wurde in den Experimenten als Gaußverteilung mit $\sigma = 3$ gewählt, der Abstand vom Mittelpunkt der Gaußverteilung entsprach gerade dem Abstand zwischen den Positionen der Worte bei einem linearen Alignment, als Zusammenhangsmaß diente die *mutual information* (siehe Abschnitt 2.6). Mit diesem einfachen Modell sind bereits auf einem Corpus, der so klein ist wie der hier zur Verfügung stehende, viele der in der Literatur bekannte Resultate erklärbar (siehe Abbildung 4.3 und vergleiche dazu die Resultate von [KR93]). Auffällig an allen diesen Listen ist, daß es eine große Tendez gibt, Worte zuzordnen, die keine Übersetzungen voneinander sind, eine korrekte Übersetzung des einen Wortes aber oft in der Nähe des anderen Wortes vorkommt. Die Experimente können mit dem PERL -Skript *iltcross* durchgeführt werden, daß noch spezielle Funktionen für die Handhabung von Interlinguatexten beinhaltet.

Erstellung von Wortfeldern Selbst wenn man nur eine recht ungenaue Schätzung hat, welches Wort die Übersetzung eines Wortes in einem gegebenen Text ist, kann man mehrsprachige (hier immer zweisprachige) Klassen von Worten bilden. Dazu kann man bei dem Ansatz zur Wortklassifikation aus Kapitel 3 die Definition 3.4 so erweitern, daß der Kontext eines Wortes gegeben ist durch die Worte aus dem eigentlichen Text *und* den Worten aus dem Parallelcorpus, die in Übersetzungspositionen stehen. Wenn man dann eine entsprechendes Zählverfahren in die Funktion t und θ einbringt, könnte man die Worte aus zwei Sprachen parallel klassifzieren. Die reicht allein jedoch nicht aus: Die vorgestellten Methoden liefern eine Klassifikation, die in unserer Domäne mühelos die deutschen und englischen Worte trennt. Man muß einen Schritt weitergehen und nicht nur den Kontext eines Wortes aus Worten beider Sprachen bestehen lassen, sondern auch das englische Wort an seiner Position im deutschen Satz eintragen und diesen Kontext mit hinzurechnen (siehe Abbildung 4.4).

Einen Ansatz mit der Intention, solche Wortfelder zu erstellen wurde in [Sch93d] verfolgt und ebenfalls auf den großen Hansards durchgeführt. Soweit man dies an seinen Abbildungen beurteilen kann unterscheidet sich die Güte der hier vorgestellten Analyse kaum von der dort erzielten, obwohl dieser auf wesentlichen größeren Parallelcorpora (den kanadischen Parlamentprotokollen) gearbeitet hatte. In seinen Abbildungen sind stets nur die guten Beispiele aufgeführt während für die Abbildung 4.4 keine Selektion nach Güte vorgenommen wurde.

Sequenzenorientiertes Alignment Das in dieser Arbeit verfolgte sequenzenorientierte Alignment ist eine Erweiterung des oben vorgestellten Wortmodelles: Statt aber ein lineares Alignment für den ganzen Satz zu postulieren wird dieses Alignment nur als stückweise linear angenommen. Daraus ergibt sich die Verpflichtung, die Grenzen dieser Teilstücke zu bestimmen (Sequenzenbestimmung) und zuzordnen (Sequenzenzuordnung). Man kann dies zu einem iterativen Verfahren erweitern, indem man das letzte Alignment in die Bestimmung der Sequenzen und Zuordnungen einbezieht. Als Anfangsalignment kann man beim iterierten Verfahren das lineare Alignment beider Sätze unterstellen.

Sowohl Sequenzenbestimmung als auch Sequenzenzuordnung basieren auf einer Zuordnungsgüte der einzelnen Worte in den Sequenzen. Diese Zuordnungsgüte ist das Produkt von zwei Bestandteile, der eine ist die Übersetzungsgüte der Worte, der andere ist die Wahrscheinlichkeit, daß sich die zwei Worte in einer Überset-

can
können (273) wir (297) könnten (307) uns (455) kann (1202) könnte (1834) unter (2581) damit (2787) habe (2830) vielleicht (3752) bis (3794) vereinbaren (4469) hoffentlich (4558) ...
Town
geschäftsreise (218) werde (221) sein (238) stadt (281) nicht (481) auf (502) zum (533) vom (799) bin (813) hier (876) 6 (970) geschäftlich (1009) 31 (1644) wir (1773) unterwegs (1948) und (1977) 11 (2018) ...
Free
können (273) wir (297) könnten (307) uns (455) kann (1202) könnte (1834) unter (2581) damit (2787) habe (2830) vielleicht (3752) bis (3794) vereinbaren (4469) hoffentlich (4558) den (6094) bleiben (6143) ...
Saturday
samstag (50) 4 (1177) den (1982) 2 (2428) am (2641) einverstanden (7187) das (7670) samstag- (8159) mit (8449) denken (9158) verschieben (10883) ...
Time
zeit (185) hätten (829) genügend (830) wieviel (1098) termin (1311) beliebig (1447) hätte (1611) bis (1635) etwas (2078) irgendwann (2997) ...
To
bis (43) um (71) von (82) am (402) zehn (407) drei (494) fünf (514) neun (518) vier (576) eins (623) sollten (634) gerne (784) dienstag (822) ich (867) den (914) zwölf (920) morgen (1030) halb (1037) wäre (1093) zwei (1119) ...
Sorry
tut (123) leid (151) mir (234) entschuldigung (1602) nein (1660) aber (2536) furchtbar (6968) am (7684) seht (9924) meinte (10057) ...
Perhaps
vielleicht (491) geschaut (5202) einplanen (6881) schieben (8986) ende- (10640) richtig (11316) november (12640) ich (14137) ersten (14249) gegen (14992) 23 (16524) sollten (18009) möglich (19573) ...
I'll
dann (2789) ungefähr (3604) müßte (3682) es (4869) eng (5118) dauern (9227) abend (12643) etwas (13203) besser (13422) art (14260) somit (16026) wird (16243) ... lange (32653) ich (33315) sie (33391) bei (36850) ...

Abbildung 4.3: **Wörterbucherstellung:** Es wurde eine Auswahl von Worten getroffen (siehe auch Abbildung 3.4) und deren Übersetzungspartnern nach dem Rang der Übersetzunsgüte geordnet (der Rang ist in Klammern angefügt). Man sieht hier deutlich, daß systematische Verwechslungen gemacht werden. Beim Wort *Town*, daß fast ausschließlich in der Floskel *to be out of town* verwendet wurde, werden Zuordnungen zur Übersetzung der Floskel *auf Geschäftsreise sein*, *geschäftlich unterwegs sein* gemacht. Beim Wort *I'll* schließlich findet sich zu Beginn der Liste gar keine korrekt Übersetzung, da die Übersetzung *werde ich* in der Regel an einer anderen Stelle im Satz auftaucht. Aus dieser Abbildung wird neben dem Einfluß der Floskeln vor allem deutlich, daß es keine offensichtliche Heuristik gibt, aus der Liste der Übersetzungspartner eine zuverlässige Teilmenge abzuleiten.
Eine Liste, die der Reihe nach alle Übersetzungspaare enthält, ist für die ersten paar hundert Paare bis auf wenige Verwechslungen aufgrund von Floskeln korrekt – eine solche Darstellung ist aber im Gegensatz zur Abbildung oben verwirrend und hat für die Entwicklung eines Übersetzungssystems keine Relevanz.

okay-we fei-fei wir-we sollten-we uns-we irgendwann-sometime
in-in den-the nächsten-next zwei-two wochen-weeks treffen-weeks
wie-what sieht-does denn-your ihre-your terminplanung-schedule aus-schedule

all-wir right-fei fei-fei so-wir we-wir should-sollten get-uns together-uns
sometime-irgendwann in-in the-den next-nächsten two-zwei weeks-wochen
what-wie does-sieht your-ihre schedule-terminplanung look like

top lustigen scheduling	dadurch vielen dank	ausmacht darauf microphone_noise	weiters another zweistundiges	what geeigneten shoot
klar beides	obwohl wide friday	part mitbekommen verstanden	17 seventeenth konferenz	eng tight verplant
ya dunno appointment	want muste rather	we uns wir	programm beendet have	i im ich
konnten will can	direkt therell unten	minute somebody jemandem	sec bitte augenblick	den der the
porter gebaude hall	22 mai sondern	didnt anbetracht given	slot every freie	wrong jeden everyday
am on bis	anbelangt believe keinem	at to um	weiterzukommen suggest reden	left while dazu
agreeable seiten mutually	used anstehende gewohnlich	einschliesslich including spaten	must irgendwie swing	m p dauert
sowohl seminaren nachster	discussing dabei reschedule	gutes beer bier	extra discussion cover	wollte wanted vergessen
own reihe hold	yes afraid ungluecklicherweise	lang naturlich januarwoche	reicht wasnt dribbeln	mention dir runs
repeat uberlegen wiederholen	gesprache durchgehen calls sorgen	arbeit someone bail dich	mark erwarte trage begin_paper_rustle	two blocks verfugung stuck
hi hallo ann hey	ubernachste bet suggestion entirely	ausreichend big last gabe	order pizza irgendwo bestellen	verschieben shall einplanen dreh
pay yup aufpassen attention	dc aufhalten end_paper_rustle durchwachsen	b t raum deutschen	live hoffe uberleben hope	terrible ziemliches messy book
she nailed wally soweit	guter irgendein einzigste particular	bestimmten specific gedacht welcher	- betrifft regarding ortlichkeiten	diskutieren ausfuhrlicher further dies
kinder spiel playing kids	nachmittagen allen freien almost	move gee how steht	vernunftig andrea excellent reasonable	kalorien calories geradezu cream
dicht amount empty neither anzubieten	lief zusammensuchen einiges arrange unterhalten	backup spas ausweichzeit eventuelle ausweichtermin	unterbringen mittagessens gemeinsam brauchten verabreden	zerstuckelt somewhat durchgehend solid fragmented
assume anvisieren cut happen genannten	extension zimmernummer lautet durchwahl nummer	information weitere sachen informationen letztes	unternehmen kaffee coffee trinken fruhstuck	opposite square gegenteil station confused
horrible ah genauso arent tuesdays	weder thatd bestenfalls clear_throat the	seem organisieren abschlus avoid help	grosartig wars soll past fein	man wars paper notizen notes
12 itd twelfth however meiste	serious hut new bezieht neuen	thai gesprach wraps erfolgreich einzelheiten	vernunftigen hinweg restrictions vorschlag einzuwenden allied	a irgendwas starten coca cola
unmoglich means thank forward freue fast	scot scot general great night dates	etwas computer depends hangt montagen printouts	luck cancelled wenig taken zufall aussieht	hmm went cindy dana froh glad
bringe bezahlen insult stressige bearbeitet heim	maze rechts follow signs runter eating	sekretar telefonnummer secretarys mail- adresse girl	siam weit sobald kitchen siam kitchen	grostenteils impossible barely verabredungen mostly starts
i paper_ruffle far geschickt mean ubel	notwendigen nebenan einnehmen wollten gewesen door	c e ask ece ahlen fragen	read englisch etc falsch abhandlung dokument	put rush occasions schnell versuche genug
diskussion andert weiterfuhren- disturbing whether aufeinander niemanden	hatte meant thought nehme meinte eben sprach	mitten or darauffolgende september blick leerlauf middle	klappen lauft moment treffe keiner schedule hinauslaufen	box voice enter philosophie untergeschos meantime zwischenzeit
chinese working zeitpunkt restaurant total dienstags chinesische	irgendeinem do lemme you liegt would skipping	brauchte planen ruhe preferably promising herauszukommen schieben	post beantworten narrow sonne sunshine wrote answer	mine address along decide vegetarier vegetarian overtime

Abbildung 4.4: **Zweisprachige Wortfelder :** Zur Erstellung eines zweisprachigen Corpus wird mit dem Programm *interleave* ein ineinandergeschachtelter Corpus erzeugt (oben), der direkt mit dem Skript *cluster* wie ein normaler Corpus verarbeitet werden kann. Die Klassifikation unten wurde mit 1000 Klassen und einem $[-3,-1][-1,-1][1,1][1,3]$ Kontext abgeleitet, die 20 größten Klassen wurden weggelassen und ebenfalls Klassen mit ein oder zwei Elementen (Klassen mit zwei Elementen enthalten fast immer Worte mit ihren korrekten Übersetzungen).

zungsposition befinden. Die Übersetzungsgüte kann man leicht bezüglich eines Alignments ableiten, wenn man alle Wahrscheinlichkeiten dafür kennt, daß sich zwei Worte in Übersetzungspostionen befinden (siehe *Einfaches Modell zur Erstellung von Wörterbüchern* oben). Es muß also noch spezifiziert werden, wie die Wahrscheinlichkeit, sich in einer Übersetzungsposition zu befinden, bestimmt wird.

Dieses Modell erweitert das einfache Modell von oben relativ geradlinig, indem es für zwei einander zugeordneten Sequenzen ein lineares Alignment mit $\sigma = \frac{\text{Laenge der Sequenz}}{4}$ unterstellt. Damit ein bestehendes Alignment nicht nur verfeinert werden kann, darf die Wahrscheinlichkeit dafür, daß zwei Worte aus einander nicht zugeordneten Sequenzen in einer Übersetzungsposition stehen, nicht Null sein. Statt davon auszugehen, daß die gegebenen "harte" Zuordnungen sicher sind, wird die Unsicherheit ähnlich wie beim linearen Alignment durch eine Gaußverteilung der Wahrscheinlichkeit modelliert, daß eine Sequenz durch eine Zuordnung tatsächlich zugeordnet wurde. Bei den Zuordnungen, die nicht "hart" sind, wird nicht von einem linearen Alignment zwischen den Worten ausgegangen sondern angenommen, daß jede Zuordnung zwischen den Worten in den jeweiligen Sequenzen gleichwahrscheinlich ist.

Mit diesen Zuordnungsgüten der Worte soll nun eine Sequenzenbestimmung und anschließend eine Sequenzenzuordnung erfolgen. Die Sequenzenbestimmung sollte sinnvollerweise so erfolgen, daß die danach zugeordneten Sequenzen möglichst gut durch ein lineares Alignment zugeordnet werden können. Dazu wird eine durchschnittliche Zuordnungsrichtung für jedes Wort berechnet, d.h. eine mit den Zuordnungsgüten gewichtete Summe des Verhältnisses $\frac{i}{j}$, wobei i die Position des betrachteten Wortes und j die des möglicherweise zugeordneten Wortes darstellt. Die Sequenzen kann man nun bestimmen, indem man die Sequenzengrenzen an Orten starker Änderung der durchschnittlichen Zuordnungsrichtung wählt. Es hat sich bei diesem Verfahren bewährt, die Zuordnungsrichtungen zunächst mit einem Tiefpass zu filtern und danach nach lokalen Maxima der Änderungen zu suchen.

Vor der Berechnung der Zuordnung der Sequenzen wird eine Sequenzenzuordnungsgüte auf der Basis der Zuordnungsgüten der Worte in der Sequenz unter der Unterstellung eines linearen Alignments berechnet. Die Zuordnung erfolgt mit der besten Zuordnung beginnend. Falls eine Zuordnung nicht erfolgen kann, weil eine der beiden Sequenzen bereits zugeordnet wurde, dann wird diese Zuordnung verworfen – es sei denn, diese Zuordnung ist "benachbart" zu der Zuordnung, die diesen Fehler verursacht hat und man kann dieses Problem durch das Verschmelzen von zwei Sequenzen eliminieren. Die Zulassung von Verschmelzungen erhöht die Robustheit der Zuordnung gegenüber einer schlechten Sequenzenbildung.

Klassenbasiertes Modell der Wortübersetzung Aufgrund des sehr kleinen Corpus bestand die Notwendigkeit, die Anzahl der Parameter zu reduzieren. Dazu sollte ein klassenbasiertes Modell geschaffen werden. Es ist jedoch problematisch, einfach die Klassen der monolingualen Corpusanalyse zu verwenden: Nehmen wir an, in beiden Sprachen sind Klassen für die Wochentage gefunden worden. Dann kann sehr gut geschätzt werden, daß ein Wochentag statistisch gesehen nur sehr selten in ein Hilfsverben übersetzt wird, wir haben aber die Information verloren, daß *Dienstag* in *tuesday* übersetzt wird und nicht in *saturday* – es ist also negative Information gewonnen worden, aber positive Information über das Vorhandensein einer Übersetzung verlorengegangen. Die Verwendung der Wortfelder kann diesen Mangel beheben helfen: Von zwei Worten wird die Übersetzungsgüte auf der Basis der Klassenzugehörigkeit geschätzt, wenn sie zu verschiedenen Klassen gehören, wenn sie aber zur gleichen Klasse gehören, werden sie auf Basis der Einzelworte geschätzt. Durch die Wahl der Anzahl der Klassen kann der Einfluß dieses Verfahrens reguliert werden, für sehr viele Klassen ergibt nahezu das ursprüngliche

Verfahren ohne Klasse, je weniger Klassen verwendet werden, desto stärker ist der Einfluß (Ausnahme: Es wird nur eine einzige Klasse verwendet).

Besprechung des neuen Verfahrens Welche Vorteile und Nachteile ergeben sich gegenüber dem Verfahren von [BPPM93a] ?

+ geringe Anzahl von Parametern durch den Einsatz der Klassen

+ größere Störungsunempfindlichkeit gegenüber Floskeln, da diese als Sequenzen zusammen übertragen werden und zum Teil durch das klassenbasierte Verfahren absorbiert werden

+ bessere Modellierung weitreichender Verschiebung von Satzteilen, da es keine Verschiebungswahrscheinlichkeiten gibt, die dies unterbinden würden

- keine Phase außer dem Initialalignment ohne Viterbi-Alignment

- keine Formulierung als EM-Algorithmus, daher keine exakte statistische Deutung

Trotz dieser Vorteile, die sich insbesondere für unsere Anwendung ergaben, war ein Alignment, daß ein Training einer angeschlossenen Transferkomponente ermöglicht hätte, nicht möglich. Über die Gründe dafür kann man ohne eine detailierte und umfangreiche Untersuchung nur spekulieren. Einige Phänomene waren jedoch sehr auffällig und sollen daher angesprochen werden:

- Trotz des klassenbasierten Ansatzes wurden die Übersetzungsgüten für schlecht modellierte Worte kaum verbessert – unter diesen schlecht modellierten Worten befinden sich insbesondere solche, die nicht in den Übersetzungspositionen eines linearen Alignments vorhanden waren (besonders klares Beispiel sind separierte Prefixe wie bei *fest-machen* und *an-melden*

- Echte Konstituentenverschiebungen zum linearen Alignment wurden nur dann vorgenommen, wenn einzelne Worte besonders sicher modelliert waren (im Extremfall nur in dieser Äußerung vorkamen) und fanden dann meist nur über kurze Entfernungen statt – es fand also keine grundlegende Revision und Präzisierung des linearen Alignments statt.

- Die mit der einsprachigen Corpusanalyse gefundenen Sequenzen waren beim Vergleich auf den zwei Corpora in der Regel nicht kompatible, das heißt, daß kaum eine Möglichkeit der direkten Zuordnung von Sequenzen gegeben war (siehe Abbildung 4.6). Bei der Anwendung des iterierten Verfahrens werden zwar oft manuell zuordenbare Sequenzen gefunden, es ergibt sich aber das Problem, daß ein maschinelles Verfahren nun nicht nur identifizieren muß, welche Sequenz welcher anderen zugeordnet werden soll sondern auch bestimmen muß, auf welcher Zusammenfassungsebene die Sequenzen jeweils zugeordnet werden sollten. Solche Algorithmen für ein strukturelles Aligment wie z.B. [MIU93] wurden versuchsweise auch von Ye-Yi Wang (persönliche Kommunikation) auf automatisch entwickelte Strukturen angwendet, was allerdings ebenfalls nicht zum Erfolg führte.

4.2 Statistische Transfermethoden

Die in der Literaur bekannten Transferverfahren lassen sich in zwei Gruppen einteilen, in solche zur Bestimmung der Übersetzung eines einzelnen Wortes und solche zur Umordnung des Satzbaus.

Das primitivste denkbare Modell ist ein solches, daß immer die beste Übersetzung jedes einzelnen Wortes einsetzt und keine Umordnungen des Satzes vornimmt. Die so enstehenden "Übersetzungen" sehen zwar schrecklich aus (siehe Abbildung 4.5), können aber als erste Referenz für ein Übersetzungssystem dienen, da man an der "Übersetzung" ungefähr erkennen kann, was der Sprecher intendiert. [BCP+90] schlägt vor, solche Sequenzen auf der Basis eines Trigrammmodelles umzuordnen.

Als erste Verbesserung wird in der Regel die Wortselektion verbessert. Dies geschieht mit Lernverfahren, die aus konkreten Alignments die Wortselektion erlernen. Zu diesem Zweck können praktisch beliebige Lernverfahren eingesetzt werden, darunter sind sehr einfache mit größerem Kontext und sehr großem Rechenaufwand wie die beim EBMT [FI92] verwendeten oder aber Entscheidungsbäume auf einem schmalen Kontext [BPPM91]. Besonders interessant wäre es gewesen, Methoden die üblicherweise beim *Tagging* benutzt werden, auf die Wortselektion auszudehnen. Das Taggingproblem ist das Problem, einem Wort in einem Text seine Wortklasse zuzuordnen. Statt einem Wort seine Wortklasse zuzuordnen kann man ihm die korrekte Übersetzung zuordnen. *hidden-markov* Modelle lassen sich auf diese Aufgabe nicht erweitern, da die Namen und der Sinn der zugeordneten Wortklassen sich nicht von Wort zu Wort ändern lassen. Anders sieht dies bei dem *transformation based learning* nach [Bri93][Bri92] aus, daß auch sehr erfolgreich auf das Tagging-Problem angewendet wurde und zu dem es eine frei verfügbare Implementierung gibt. Dieses Verfahren, daß symbolische Regeln lernt, die aufgrund des Kontextes das Problem entscheiden, hätte direkt angewendet werden können, da es für jedes Wort einen eigenen Regelsatz hält und damit auch das Ausgabevokabular für jedes Wort unabhängig gewählt werden kann.

Unter dem Oberbegriff TDMT (*transfer driven machine translation*) wird ein Bündel von Methoden zusammengefaßt, daß schrittweise an bestimmten Übersetzungsproblemen orientiert aufgebaut wurde. Das Problem bei diesem System für die Zielrichtung dieser Arbeit war es allerdings, daß die Methoden wesentlich von der manuellen Spezifikation abhängen, *was* als Übersetzungsproblem zu verstehen und zu lösen ist. Im Gegensatz dazu könnte ein System stehen, daß mithilfe von Algorithmen des *transformation based learning* eine möglicherweise strukturierte Repräsentation schrittweise umordnet und transformiert. In [Bri93] werden Methoden beschrieben, um die Klammerstruktur von Sätzen aus Beispielen zu lernen und die so erhaltene Struktur anschließend mit Markierungen zu versehen. Das *transformation based learning* ist eine einfache und sichere Methode, die man mit hoher Wahrscheinlichkeit auch auf das zu diesen Problemen analoge Problem der Umwandlung und Umstellung von einer Sprache in die andere benutzen kann. Das einzige, was man dazu wissen muß, ist, wie man bewertet, daß eine Teiltransformation "erfolgreich" war in dem Sinne, daß sie der Lösung, daß heißt der Übersetzung, näher gekommen ist. Dies sollte insbesondere dann gut funktionieren, wenn man weiß, welche Teile aus den Spezifikationen einander zuzuordnen sind. Dieses Problem ist das bereits oben angesprochene strukturelle Alignment [MIU93].

4.3 Alignmentfreie Übersetzungsmethoden

Nachdem oben besprochen worden ist, daß das Alignment bei der Erstellung des Übersetzungssystems eine entscheidende Hürde darstellte muß man weiterfragen, welche Methoden zur Übersetzung ohne ein Alignment in Frage kommen. In der Literatur ist dazu nur die holistische Übersetzung bekannt, die die Übersetzung von einer in die andere Sprache in einem Schritt durchführt (daher holistisch). [Chr91] stellt eine Implementierung dieses Übersetzungsverfahrens vor, daß auf der konfluenten Analyse mit neuronalen Netzen basiert (siehe Abschnitt 2.3.2). Die internen

Repräsentationen der neuronalen Netze lassen sich in gewisser Weise als Interlingua betrachten, die aus dem Problem automatisch entwickelt werden und deren Entwicklung auch gleichzeitig die Erstellung eines Systems zur Übersetzung beinhaltet. Die in [Chr91] sehr eingegrenzten Trainingssätze, die außerdem verglichen mit den Längen der Sätze bei den Terminabsprachedialogen nur sehr kurz sind, läßt allerdings einen Erfolg als fraglich erscheinen. Die Experimente zur Bestimmung von Sprechakten ließen außerdem auch deutlich werden, daß die konfluente Analyse zwar recht gut funktioniert aber auch nicht beliebig komplexe Strukturen abspeichern kann (siehe Kapitel 3.4).

Die Idee, daß bei der konfluenten Analyse eine Art Interlingua gelernt wird, kann zu der Idee führen, eine strukturierte Interlingua automatisch zu lernen und so das Alignmentproblem abzuschwächen oder gar zu eliminieren. Die Methoden zum unüberwachten Erlernen strukturierter Repräsentationen, die im Bereich des maschinellen Lernens bekannt sind [Seg90][CH94], scheinen derzeit aber selbst mit einfachen Aufgaben aus dieser Domäne weit überfordert zu sein (siehe Kapitel 3.3.2, Abschnitt *Das Minimum Description Length Principle* zu einem Experiment mit der Methode von [CH94]).

ganzen ordnung uns somit wir sollten uns einmal irgendwann in den nächsten zwei wochen was sieht ihrem kalender aus als
schauen mal nächsten dienstage gut es ginge habe bis werde in den nachmittag oder oder donnerstag unmöglich gut NO eher gut bei sie
unglücklicherweise dienstag dienstag zum donnerstag der nächsten woche werde werde werde der stadt sind beliebig anderen tage nächsten woche prima oder vielleicht wir sollten immer bis den 2 woche
also wie sieht montag den acht aus bis sie
am montag ich regelmäßige ausgebucht von neun erst zwei und es sieht als mein drei uhr meeting war abgesagt somit lunch zwischen zwei bis sechs ich können wahrscheinlich uns einverstanden sie
also vielleicht wir sollten immer bis den nächsten woche dann wie sieht dienstag mittwoch oder donnerstag aus bei sie den nachmittagen sind sehr gut bei mir und mittwoch ist zeit ganzen tag

Abbildung 4.5: **Pigeon Deutsch:** Die Verwendung der automatisch erzeugten Wörtebüchern führt zwar zu Resultaten einer kaum akzeptablen Qualität, man kann an diesem Resultat aber bereits erkennen, was man für eine verständliche Übersetzung vor allem braucht: Eine bessere Übersetzung des einzelnen Wortes und eine Behandlung von Floskeln. Es ist aber so, daß eine ähnliche Übersetzung mit einem "richtigen" Wörterbuch zu eher unverständlicheren Resultaten führt. Die Verschleifung der Bedeutungen führt also dazu, daß die Übertragung zunächst verbessert wird, da eine domänenspezifische Übersetzungsvariante gewählt wird. Eine Umordnung der Worte im Satz erscheint dagegen eher unwichtig oder gar nicht erforderlich.

Abbildung 4.6: **Anwendung der einsprachigen Corpusanalyse zur Übersetzung:** Die unüberwachte Corpusanalyse wurde auch mit der Hoffnung verfolgt, hiermit das Zuordungsproblem vereinfachen zu können. Diese Abbildung zeigt zwei typische Sätze, bei denen Probleme auftreten (Darstellung wie in Abbildung 3.8). Das erste auffällige Problem liegt in der einsprachigen Corpusanalyse, die nur auf dem kleinen Corpus der tatsächlich übersetzten Sätze durchgeführt wurde. Daher ergibt sich im Englischen das Problem, daß die Sequenz *are any* falsch an *out of town* adjungiert wurde. Weitere Probleme liegen in der Struktur der beiden Sprachen, wie man am Beispiel der Übersetzung von *I'll* oder gar von *are ... fine*, daß in *ginge es denn (noch)* übersetzt wird.

Kapitel 5

Schlußfolgerung

Diese Diplomarbeit hat sich vor allem mit Methoden zur statistischen und empirischen Analyse einsprachiger Corpora beschäftigt. Dabei war von vorneherein die Aufgabenstellung so gewählt, daß das Ziel die Analyse und Übersetzung spontan gesprochener Sprache war. Diese Voraussetzungen implizieren eine kleine Datenmenge mit einer hohen Variabilität, für die es darüberhinaus wenige andere Methoden gibt, um eine Analyse zu erstellen. In dieser Arbeit wurde auf einer ganzen Reihe von Feldern parallel gearbeitet, die unten dargestellt werden sollen.

Einsprachige Corpusanalyse
Eine unüberwachte Analyse einsprachiger Corpora ist durch eine gute Modellierung auch bei kleinen und verrauschten Texten mit Erfolg möglich. Erfolgreich heißt hier, daß die so erzeugten Modelle sowohl durch komplexitäts- wie informationstheoretische Maße nachweislich eine hohe Güte besitzen, als auch durch die erfolgreiche Anwendung zur Sprechakt- und Dialogmodellierung. Die hier erstellten Modelle leisten eine unüberwachte, hierarchische Analyse des Textes, die durch andere Methoden bisher noch nicht in dieser Form möglich ist. Das nächste Ziel auf diesem Gebiet muß in der Erschließung weiterer Anwendungen dieser Technik liegen, die nicht nur auf der Wortebene ansiedeln müssen – die Vision von [Pow89], ein auf wenigen Grundprinzipien basierendes System zu erstellen, daß von der Signal- bis zur Verständisebene menschliche Sprache unüberwacht erlernt, würde damit näher rücken.
Als erstes großes Themengebiet soll die Anwendung der Corpusanalyse auf unterschiedliche Ebenen von Sprache angesprochen werden. Die hier untersuchte Ebene ist die Wortebene und es wurde versucht, von dieser Ebene eine Analyse auf Satz- bzw. Sprechaktebene durchzuführen. Eine andere Ebene ist die Phonemebene, für die ein kurzes Resultat in Abbildung 3.3 gegeben wurde. Ein nicht dokumentiertes Experiment auf der Basis der *senones* in der *resource management* Datenbasis hat gezeigt, daß die hier entwickelten Methoden zur Sequenzensuche sich auch auf die phonetische Ebene übertragen lassen. Auf diese Weise ließen sich morphologische Muster ableiten, die zur Erstellung von Aussprachewörterbüchern geeignet sind. Schließlich kann diese Analyse auch direkt auf die Signalebene angewendet werden. Eine andere interessante Untersuchung wäre es, auf der Basis von Sprechaktrepräsentationen strukturierte Verläufe von Dialogen abzuleiten. Als initiale Sprechaktrepräsentation könnte eine Analyse des Sprechaktes mit den hier vorgestellten Methoden zur Corpusanalyse dienen.

Das zweite Gebiet ist die Verbesserung der Wortklassifikation durch zusätzliche Informationen. Die Güte der Wortklassifikation hat sich zwar für die Anwendung statistischer Verfahren bewährt, aber es gibt noch Möglichkeiten, sie zu verbessern. Die entscheidende Möglichkeit zur direkten Änderung des Verfahrens ist eine Änderung des berücksichtigen Kontextes, der in die Klassifikation einfließt. Dazu kann

der jetzt lokale Kontext erweitert werden, um weitreichendere Informationen, die entweder durch die Sequenzenanalyse oder aber durch einen Parser geliefert wurden.

Das dritte Gebiet ist der Einsatz dieser Analysen zur Verbesserung der Eingaben in Parser. Durch die vorgenommenen statisische Analyse findet in gewisser Weise ein Vorverarbeitung des Textes statt, die von einem Parser genutzt werden könnte. Dies trifft insbesondere für die in der Gruppe entwickelten neuronalen Parser zu, für die derzeit nur von Hand entwickelte Repräsentationen zur Verfügung stehen. Durch die einsprachige Corpusanalyse können Worte klassifiziert werden und diese Klassen als Eingabereräsentation verwendet werden. Der Einsatz der Fuzzy-Klassifikation erlaubt die Erstellug von Eingaberepräsentationen, die keine n aus m Kodierungen darstellen (siehe auch Abbildung 3.6). Zusätzlich zu den Wortklassifikationen können Repräsentationen der aufgrund der Analyse gefundenen Sequenzen benutzt werden, um die Darstellung des Kontextes eines Wortes zu verbessern.

Das vierte Gebiet ist die konkrete Implementierung und der Test der hier vorgestellten Methoden zur Sprachmodellierung. Da diese Methoden sich deutlich von den bekannten Modellen unterscheiden und die iterierte Corpusanalyse ein sehr präzisieses Bild von Sprache abgibt ist es ungeklärt, ob diese Methoden sich zum Einsatz in einem Spracherkenner eignen und möglicherweise traditionelle Methoden übertreffen oder nicht.

Das fünfte Gebiet stellt die Behandlung von Worten dar, die in der Trainingsmenge nicht vorgekommen sind. Diese erhalten in der derzeitigen Implementierung keine Klasse zugeordnet und führen daher dazu, daß sie nicht in eine Sequenz eingebunden werden können. Dies kann durch Verfahren zur Klassifikation unbekannter Worte, wie in [Bri93] eliminiert werden.

Alignment

Die automatische Analyse und das Zuordnungsproblem zweisprachiger Corpora sind weiterhin offen. Es wurde gezeigt, daß man bereits mit trivialen Ansätzen Wörterbücher erstellen kann, deren direkter Einsatz bereits verstehbare "Übersetzungen" erzeugt. Ein echte Verbesserung kann aber bei dem vorliegenden spontansprachlichen Corpus nur dann erfolgen, wenn die Floskeln automatisch erkannt und zugeordnet werden können. Dazu wurden eine Reihe von Verbesserungen von Standardverfahren getestet und die Verbindung zur einsprachigen Analyse hergestellt, was noch nicht ausreichte, um das Problem erfolgreich zu bewältigen. Die Einführung weiteren Wissens bei der Lösung des Aligmentproblems ist nach Meinung des Autors unerläßlich und noch wichtiger als die Erweiterung der Trainingsdatenmenge.

Wie im Kapitel 4 erläutert wird, ist eine Zuordnung rein auf Wortbasis offensichtlich sehr schwierig, da der Einfluß von feststehenden Floskeln eine Statistik auf der Basis von Wortzuordnungen erheblich erschwert. Als Alternative dazu bietet sich eine Zuordnung auf der Basis von bereits analysierten Sätzen an. Trotz der angewendeten Corpusanalyse scheint es auch nicht möglich zu sein, eine Zuordnung auf der Basis einer *beliebigen* hierarchischen Organisation des Textes (z.B. der durch eine unüberwachte Methode erzeugten) zu erstellen. Dies liegt vor allem daran, daß die Statistik der Sequenzen, von denen es sehr viel weniger Beispiele als von den Worten gibt, noch schlechter ist. Die Konsequenz daraus ist, daß man besondere Formen hierarchischer Analyse betrachten muß, nämlich solche, bei denen die Zuordnung besonders einfach ist. Dazu bietet es sich an, beide Korpora durch eine *wechselseitig konsistente* Methode zur Identifikation von Sequenzen vorzuanalysieren, sodaß die Zuordnung der Sequenzen und Worte lediglich eine banale Aufgabe darstellt. Besonders einfach ist diese Zuordnung insbesondere dann, wenn die Sequenzen mit *konsistenen* Bezeichnern versehen sind, sodaß die Zuordnung durch eine direkte Identifikation der Bezeichener vereinfacht werden kann. Als Verfahren für diese *wechselseitig konsistente* Identifikation von Sequenzen können zum einen

die in der Gruppe entwickelten neuronalen Parser mit einer enstprechenden Ausgaberepräsentation dienen oder aber der in der Gruppe entwickelte Pattern-Matching Parser (siehe auch Kapitel 1). Insbesondere dann, wenn beim Pattern Matching die Identifikation der Auslöser für die einzelnen Slots bekannt ist und diese konsistent sind, kann eine genaue Identifikation der Bestandteile erfolgen[1]. Die durch das Pattern Matching System erstellten flachen Analysen sind vor allem für eine Wortzuordnung geeignet. Ob dies allerdings auch zu einer Lösung des Problems der Umordnung der Satzstruktur führt, ist fraglich, da hierzu wahrscheinlich eine genauere Analyse benötigt wird.

Robuster Transfer

Aufgrund der Probleme bei der Erstellung eines Alignment wurden in dieser Arbeit keine Experimente zum Transfer durchgeführt (siehe auch Abschnitt 1.1.2 und 4.2)[2].

Die im Laufe der Arbeit durchgeführte Literaturrecherche hat jedoch ergeben, daß die Methoden nach [Bri93] die derzeit hoffnungsvollsten für das Erlernen der eigentlichen Übersetzungskomponente darstellt. Die Methoden aus dem Bereich *Example Based Transfer* (EBT) und der *Transfer Driven Machine Translation* fallen dahinter zurück, da sie zum einen immer für ein spezielles Übersetzungsproblem konstruiert werden und zum anderen die Lernverfahren in der bisher präsentierten Form dazu führen, daß die Übersetzung eines Satzes sehr aufwendig ist.

In dieser Arbeit konnte ebenfalls nicht die Technik der holistischen Übersetzung (siehe Kapitel 4.3) getestet werden, die zwar sicher keine Allzweckmethode zur Übersetzung ist, sich aber möglicherweise in speziellen Übersetzungssituationen eignet. So könnte z.B. der Transfer von identifizierten Abschnitten einer Phrase durch ein spezielles Netz erfolgen. Diese Methode ist auch deshalb erfolgversprechend, weil die dabei verwendeten RAAM-Netze bei den in dieser Arbeit mit neuronalen Netzen durchgeführten Experimenten sehr gute Resultate brachten und sehr einfach trainierbar waren.

Über diese Arbeit

Diese Arbeit ist in ihrer Zielsetzung bereits sehr ambitioniert gewesen. Einen vollautomatischen Übersetzer für spontane Sprache aus 1000 Beispielsätzen ohne menschliches Eingreifen zu erstellen, ist aber auch hier nicht gelungen. Aber diese Arbeit führt vor, daß sehr viel mehr vollautomatisch geht, als man sich zunächst vorstellen mag und zeigt die Grenzen auf, die derzeit bestehen. Eine Einsicht, die für den Erfolg dieser Arbeit entscheidend war und noch vielmehr über den Erfolg zukünftiger sein wird, ist, daß die Entwicklung von Methoden, die auf statistischer Basis Texte analysieren, von einer kreativen Integration linguistischer *und* statistischer Kriterien geprägt sein sollte. Bei der Verarbeitung der hierbei entstehenden Datenmengen muß zudem eine große Sorgfalt bei der Implementierung geleistet werden, die bei diesen Verfahren entscheidend sein kann. Da sich diese Forschungsrichtung erst als eigenständiger Zweig konstituiert, sind solche Arbeiten noch relativ rar und es soll hier vor allem auf die positiven Beispiele Suhotin, Brill, Finch und Schütze verwiesen werden.

Diese Gründe mögen zum Teil erklären, warum im Jahre 1994 diese Arbeit noch zu leisten war. Der andere Grund, eine Arbeit wie die hier vorliegende gerade heute zu machen, ist, daß die Rechenleistungen und die maschinenlesbaren Corpora erst in den letzten 10 Jahren verfügbar wurden, mit denen Analysen wie die hier gezeigten möglich sind. Diese Ausgangslage wird sich aufgrund der absehbaren Entwicklung in den nächsten Jahren noch erheblich verbessern.

[1] Zum Zeitpunkt der Erstellung dieser Arbeit waren die technischen Voraussetzungen für ein solches Vorgehen noch nicht erfüllt, es war z.B. die deutsche Grammatik für den Pattern Matching Parser erst in der Entwicklung.

[2] Es hätte der Intention der Arbeit, ein nahezu vollautomatisiertes System zu erstellen, wiedersprochen, große Mengen von Trainingsdaten von Hand zu erstellen.

Literaturverzeichnis

[Abn90] Steven Abney. Rapid incremental parsing with repair. In *Proceedings of the 6th New OED Conference, University of Waterloo, Ontario*, pages 1–9, 1990.

[Abn91a] Steven Abney. Chunks and dependencies: Bringing processing evidence to bear on syntax. Draft, October 1991.

[Abn91b] Steven Abney. *Principle Based Parsing*, chapter Parsing by Chunks. Kluwer, 1991.

[Abn91c] Steven Abney. *Views on Phrase Structure*, chapter Syntactic Affixation and Performance Structure. Kluwer, 1991.

[Abn92] Steven Abney. Prosodic structure, performance structure and phrase structure. In *DARPA Speech and Natural Language Workshop, San Mateo, CA*, pages 425–428. Morgan Kaufmann, 1992.

[acl92] *Proceedings of the 30th meeting of the Association for Computational Linguistics*, 1992.

[acl93] *Proceedings of the 31th meeting of the Association for Computational Linguistics*, 1993.

[AD92] Hussein Almuallim and Thomas G. Dietterich. On learning more concepts. In *Machine Learning, Proceedings of the Ninth International Conference*, 1992.

[All90] Robert B. Allen. Connectionist language users. *Connection Science*, 2:279–311, 1990.

[ALP66] ALPAC. Languages and machines. computers in translation and linguistics. Technical report, ALPAC, 1966.

[ATR93] ATR Interpreting Telephone Company. *International Workshop for Statistical Translation*, 1993.

[Bat92] Batt. First- and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4:144–166, 1992.

[BCP+90] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jellinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[BDO+93] Michael E. Berry, Theresa Do, Gavin O'Brien, Vija Krishna, and Sowmini Varadhan. Svdpackc (version1.0) user's guide. Technical Report CS-93-194, Computer Science Department, University of Tenessee, 1993.

[BdP+92] Peter F. Brown, Peter V. deSouza, Vincent J. Della Pietra, Robert L. Mercer, and Jennifer C. Lai. Class based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[Ber91] George Berg. Learning recursive phrase structure: Combining the strenths of pdp and x-bar syntax. Technical Report TR 91-5, University at Albany, New York, 1991.

[Bez73] James C. Bezdek. *Fuzzy mathematics in pattern classification.* PhD thesis, Cornell University Ithaka NY, 1973.

[Bez92] James C. Bezdek. Computing with uncertainty. *IEEE Communications Magazine*, pages 24–36, September 1992.

[Bib92] Douglas Biber. Co-occurrence patterns among collocations: A tool for corpus-based lexical knowledge acquisition. *Computational Linguistics*, 19(531-538), 1992.

[Bla93] Ezra Black. Parsing english by computer: The state of the art. [ATR93].

[BLM91] Peter F. Brown, J. C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Proceedings 29th Annual Meeting of the Association for Computational Linguistics*, Juni 1991.

[BMS90] Eric Brill, Mitchell Marcus, and Beatrice Santorini. Deduction linguistic structure from the statistics of large corpora. In *Speech and Natural Language Workshop, San Mateo, California*, pages 272–285, 1990.

[BPP+] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, and Surya Mohanty. Dividing and conquering long sentences in a translation system.

[BPPM91] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. Word sense disambiguation using statistical methods. In *Proceedings 29th Annual Meeting of the Association for Computational Linguistics*, Juni 1991.

[BPPM93a] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation. *Computational Linguistics*, 19(1):75–102, 1993.

[BPPM93b] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1993.

[BPW94] Finn Dag Buø, Thomas Polzin, and Alex Waibel. Learning complex output representations in connectionist parsing of spoken language. In ICASSP-94 [ICA94].

[Bra91] Rüdiger Brause. *Neuronale Netze.* Teubner-Verlag, 1991.

[Bre93] Michael R. Brent. From grammar to lexicon. *Computational Linguistics*, 19(2), 1993.

[Bri92] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*, Trento, Italy, 1992.

[Bri93] E. Brill. *A Corpus-Based Approach to Language Learning.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1993.

[Buø92] Finn Dag Buø. *A learnable connectionists parser that outputs feature structures.* Phd proposal, Universität Karlsruhe, 1992.

[Bur91] Laura Ignizio Burke. Clustering characterization of adaptive resonance. *Neural Networks*, 4:485–491, 1991.

[BW70] D. M. Boulton and C.S. Wallace. A program for numerical classification. *The Computer Journal*, 13:63–69, 1970.

[BWYK91] Gautam Biswas, Jerry B. Weinberg, Qun Yang, and Glenn R. Koller. In *Machine Learning, Proceedings of the Eighth International Conference*, 1991.

[CDB86] Robert L. Cannon, Jitebdra V. Dave, and James C. Bezdek. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and machine Intelligence*, PAMI-8(2):248–255, March 1986.

[CDG93a] Kenneth Church, Ido Dagan, and William Gale. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora.* ACL, 1993.

[CDG+93b] Kenneth Church, Ido Dagan, William Gale, Pascale Fung, Jon Helfman, and Bala Satish. Aligning parallel texts: Do methods developed for english french generalize to asian languages ? In *Pacific Asia Conference on Formal and Computational Linguistics*, 1993.

[CG92] Gail A. Carpenter and Stephen Grossberg. A self-organizing network for supervised learning, recognition and prediction. *IEEE Communications Magazine*, pages 38–49, September 1992.

[CGR91a] Gail A. Carpenter, Stephen Grossberg, and John H. Reynolds. Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4:565–588, 1991.

[CGR91b] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Art 2-a: An adaptive reasoning algorithm for rapid category learning and recognition. *Neural Networks*, 4:493–504, 1991.

[CGR91c] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[CH83] Jaime G. Carbonell and Phillip J. Hayes. Recovery strategies for parsing extragrammatical language. *American Journal of Computational Linguistics*, 9(3-4):129–146, 1983.

[CH94] Diane Cook and Lawrence B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.

[Cha90] David J. Chalmers. Syntactic transformations on distributed representations. *Connection Science*, 2:53–62, 1990.

[Cha93] Geofrey J. Chappell. The temporal kohonen map. *Neural Networks*, 6:441–445, 1993.

[Chr91] Lonnie Chrisman. Learning recursive distributed representations for holistic computation. *Connection Science*, 3:345–366, 1991.

[Con93] Dennis Connolly. Constructing hidden variables in bayesian networks via conceptual clustering. In *Machine Learning, Proceedings of the Tenth International Conference*, 1993.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley-Interscience, 1991.

[Dia92] Konstantinos I. Diamantaras. *Principal Component Learning Networks and Applications*. Phd thesis, Princeton University, Department of Electrical Engineering, 1992.

[Die] Joachim Diedrich. *Steps Towards Knowledge Intensive Learning*, chapter 11, pages 284–304.

[Die89] Joachim Diedrich. Instruction and high-level learning in connectionist networks. *Connection Science*, 1(2):161–180, 1989.

[Die90] Joachim Diedrich. Recruitment vs. backpropagation learning: Relearning in connectionist networks. Arbeitspapier 457, GMD, 1990.

[Die91] Joachim Diedrich. Re-learning in connectionist semantic networks. In *IJCAI-91 Wokshop: Evaluating and Changing Representation in Machine Learning*, 1991. Sydney.

[DLR77] A. P. Dempster, N.M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977. with discussion.

[DTB93] Joachim Diedrich, Andreas Tümmel, and Eric Bartels. Recurrent and feedforward networks for human-computer interaction. [eca93], pages 206–207.

[Dun93] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[dVP92] Bert de Vries and Jose C. Principe. The gamma model – a new neural model for temporal processing. *Neural Networks*, 5:565–576, 1992.

[eca93] *European Conference on Artificial Intelligence*, 1993.

[Ell91] T Mark Ellsion. Discovering planar segregations. In Powers and Reeker [PR91], pages 42–47.

[Elm90] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[Elm91] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225, 1991.

[Fah90] Scott E. Fahlman. The recurrent cascade-correlation architecture. [nip90], pages 190–196.

[FC92] Steven Finch and Nick Chater. Bootstrapping syntactic categories using statistical methods. In *Learning of Natural Language: Proceedings of the first SHOE Workshop*, pages 230–235. Katholieke University Brabant, Holland, 1992.

[FI92] Osamo Furuse and Hitoshi Iida. Cooperation between transfer and analysis in example-based framework. In *Computational Linguistics*, 1992.

[Fin93] Steven Finch. *Finding Structure in Language*. PhD thesis, University of Edinburgh, 1993.

[FL91] Scott E. Fahlman and Christian Lebier. The cascade-correlation learning architecture. [nip91], pages 524–532.

[Fri92] Bern Fritzke. Kohonen feature maps and growing cell structures – a performance comparision. [nip92], pages 123–130.

[FW83] Dan Fass and Yorick Wilks. Preference semantics, ill-formedness, and metaphor. *American Journal of Computational Linguistics*, 9(3-4):178–187, 1983.

[Gal91] Stephen I. Gallant. A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3:293–309, 1991.

[Gas] Michael Gasser. Networks that learn phonolgy. neuropros.

[GC93] William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.

[GL90a] Michael Gasser and Chan-Do Lee. neuroprose, 1990.

[GL90b] Michael Gasser and Chan-Do Lee. A short-term memory architecture for the learning of morphophonemic rules. [nip90], pages 605–611.

[GM91] C.L. Giles and C.B. Miller. Extracting and learning an unkown grammar with recurrent neural networks. [nip91]. also on neuroprose.

[Gün91] Ralph Günther. Untersuchung von cluster-algorithmen und ihre parallelisierung. Diplomarbeit, THW Aachen, 1991.

[Gra83] Richeard H. Granger. The nomad system: Exception-based detection and correction of errors during understanding of syntactically and semantically ill-formed tex. *American Journal of Computational Linguistics*, 9(3-4):188–196, 1983.

[Gri93] Ralph Grishman. Linguistic knowledge acquisition from monolingual and bilingual corpora. [ATR93].

[Guy91a] Jacques B.M. Guy. Statistical properties of two folios of the voynich manuscript. *Cryptologica*, XV(3):207–218, 1991.

[Guy91b] Jacques B.M. Guy. Vowel identification: An old (but good) algorithm. *Cryptologica*, XV(3):258–262, 1991.

[Hal90] Brigitte K. Halford. *Syntax gesprochener Sprachen*, volume 14 of *ScriptOralia*, chapter The Complexity of Oral Syntax, pages 33–44. Tübingen: Narr, 1990.

[Har90] Mary Hare. The role of similarity in hungarian vowel harmony: a connectionist approach. *Connection Science*, 2:123–150, 1990.

[Hen92] James B. Henderson. A connectionist parser for structure unification grammar. [acl92], pages 144–151.

[Her90] Sandor Hervey. *Syntax gesprochener Sprachen*, volume 14 of *ScriptOralia*, chapter Discrepancies between "oral" and "written" Texts; The Case from a Translation Theoretical Viewpoint, pages 27–32. Tübingen: Narr, 1990.

[HH89] Günther Hämmerlein and Karl-Heinz Hoffmann. *Numerische Mathematik*. Number 7 in Grundwissen Mathematik. Springer-Verlag, 1989.

[HK92] K. Hornik and C.-M. Kuan. Covergence analysis of local extraction algorithms. *Neural Networks*, 5:229–240, 1992.

[HL93] Julia Hirschberg and Diane Littman. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530, 1993.

[HLL91] Peter M. Hastings, Steven L. Lythinen, and Robert K. Lindsay. Learning words from context. In *Machine Learning, Proceedings of the Eighth International Conference*, 1991.

[HN92] Eduard Hovy and Sergej Nirenburg. Approximating an interlingua in a principled way. DARPA, 1992.

[HP93] Christa Hauhenschild and Birte Prahl. Konzept translationsprobleme - translationsstrategien. Memo on VERBMOBIL CD 1, 1993.

[ICA94] IEEE. *ICASSP*, 1994.

[Iid93] Hitoshi Iida. Beyond analysis centered large scale mt systems. [ATR93].

[Jai91a] Ajay N. Jain. *PARSEC: A Connectionists Learning Architecture for Parsing Spoken Language*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.

[Jai91b] Ajay N. Jain. Using parsec, 1991. comes with PARSEC.

[Jaq93a] Christian Jaquemin. Activation diffusion: a connecionist network for robust parsing. [eca93], pages 183–197.

[Jaq93b] Christian Jaquemin. A coincidence detection network for spatiotemporal coding: Application to nominal composition. In *Proceedings of the WCAI 93, Chambery*, 1993.

[JHMR83] K. Jensen, G. E. Heidorn, L. A. Miller, and Y. Ravin. Parse fitting and prose fixing: Getting a hold on ill-formedness. *American Journal of Computational Linguistics*, 9(3-4):147–160, 1983.

[JJ93] Micheal I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 1993. (submitted).

[JJNH91] Robert Jacobs, Micheal I. Jordan, Steven J. Nowlan, and Geoffry Hinton. Adaptive mixture of local experts. *Neural Computation*, 3:79–87, 1991.

[Jor86] Michael I. Jordan. Serial order: A parallel distributed processing approach. ICS Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

[JWT92] Ajay N. Jain, Alex Waibel, and David S. Touretzky. Parsec: A structured connectionist parsing system. *IEEE ?*, 9(I):205–208, 1992.

[Kar90a] Lorraine F.R. Karen. Identification of topical entities in discourse: a connectionist approach to attentional mechanisms in language. *Connection Science*, 2:103–122, 1990.

[Kar90b] Fred Karlson. Constraint grammar as a framework for parsing running text. In Hans Karlgren, editor, *Papers presented to the 13th International Conference on Computational Linguistics*, volume 3, pages 168–173, 1990.

[Kaz91] Rick Kazman. *Babel:* a psychological plausible cross-linguistic model of lexical and syntactic acquisition. In *Machine Learning, Proceedings of the Eighth International Conference*, 1991.

[KF90] Stan C. Kwasny and Kanaan A. Faisal. Connectionism and determinism in a syntactic parser. *Connection Science*, 2:63–82, 1990.

[KIM91] Akira Kurematsu, Hitoshi Iida, and Tuyoshi Morimoto. Language processing in connection with speech translation at atr interpreting research laboratories. *Speech Communication*, 10:1–9, 1991.

[KN93] Reinhard Kneser and Herman Ney. Improved clustering techniques for class-based statistical language modelling. 1993.

[KR93] Martin Kay and Martin Röscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, 1993.

[Kru90] John K. Kruschke. Alcove: A connectionists model of human category leraning. [nip90], pages 649–655.

[Lan92] Trent E. Lange. Lexical and pragmatic disambiguation and reinterpretation in connectionist network. *International Journal of Man-Machine Studies*, 36:191–220, 1992.

[LD90] S. M. Lucas and R. I. Damper. Syntactic neural networks. *Connection Science*, 2:195–221, 1990.

[LD91] David LeBlanc and Henry Davis. A model for the development of phrase structure. In Powers and Reeker [PR91], pages 109–115.

[Lei92] Russel R. Leighton. *The Aspirin/MIGRAINES Neural Network Software, Users's Manual, Release V6.0.* MITRE Corporation, 1992.

[Lew90] Theodor Lewandowski. *Linguistisches Wörterbuch*, volume 1-3 of *UTB für Wissenschaft.* Quelle und Meyer, 1990.

[LG91] Philip Laird and Evan Gamble. A "pac" algorithm for making feature maps. *Machine Learning*, 6:145–160, 1991.

[Lör91] Wolfgang Lörscher. *Translation performance, translation process, and translation strategies: a psycholinguistic investigation.* Number 4 in Language in Performance. Tübingen: Narr, 1991.

[LW] Trent E. Lange and Charles M. Wharton. Remind: Retrival from episodic memory by inferenzing and disambiguation. In J. Barnden and K. Holyoak, editors, *Advances in connectionist and neural computation theory*, volume II.

[Mag92] David M. Magerman. Efficiency, robustness and accuracy in picky chart parsing. [acl92], pages 40–47.

[Mar93] Mitchell Marcus. Statistical natural language processing: Current trends and future directions. [ATR93].

[MB91] Michael C. Mozer and Jonathan Bachrach. Slug: A connectionist architecture fo inferring the structure of finite-state environments. *Machine Learning*, 7:139–160, 1991.

[MD91] Risto Miikulainen and Michael G. Dyer. Natural language processing with modular pdp networks and distributed lexicon. *Cognitive Science*, 15:343–399, 1991.

[Mii90] Risto Miikulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2:83–101, 1990.

[MIU93] Yuij Matsumoto, Hiroyuki Ishimoto, and Takehito Utsuro. Structural matching in parallel texts. In *ACL*, pages 23–30, 1993.

[MK91] Barlett W. Mel and Christof Koch. Sigma-pi learning: On radial basis functions and cortical associative learning. [nip91], pages 474–481.

[MM91] David M. Magerman and Mitchell P. Marcus. Distituent parsing and grammar induction. In Powers and Reeker [PR91], pages 122a–122e.

[Moz90] Michael C. Mozer. Discovering discrete distributed representations with iterative competitive learning. [nip90], pages 627–634.

[MS89] Michael C. Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assesment. [nip89], pages 107–115.

[Mul89] Jan Mulder. *Foundations of Axiomatic Linguistics*. The Hague, 1989.

[Nag93] Makoto Nagao. Varieties of heuristics in sentence parsing. [ATR93].

[Nea92] Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.

[NH93] Radford M. Neal and Geoffrrey E. Hinton. A new view of the em algorithm that justifies incremental and other variants. *Biometrika*, 1993. submitted.

[nip88] *Advances in Neural Information Processing Systems*, number 1, 1988.

[nip89] *Advances in Neural Information Processing Systems*, number 2, 1989.

[nip90] *Advances in Neural Information Processing Systems*, number 3, 1990.

[nip91] *Advances in Neural Information Processing Systems*, number 4, 1991.

[nip92] *Advances in Neural Information Processing Systems*, number 5, 1992.

[nip93] *Advances in Neural Information Processing Systems*, number 5, 1993.

[Now91] Steven J. Nowlan. Maximum likelihood competitive learning. [nip91], pages 574–582.

[NTS93] H. Nakanashi, I. B. Turksen, and M. Sugeno. A review and comparision of six reasoning methods. *Fuzzy Sets and Systems*, 57:257–294, 1993.

[Oja92] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[Pal80] G. Palm. On associative memory. *Biological Cybernetics*, 36:19–31, 1980.

[Paw92] Eva Pawlowski. Erweiterung von parsec um syntax-label. Studienarbeit, Universität Karlsruhe, 1992.

[Ped93] Witold Pedrycz. Fuzzy neural networks and neurocomputations. *Fuzzy Sets and Systems*, 56:1–28, 1993.

[Pil90] Herbert Pilch. *Syntax gesprochener Sprachen*, volume 14 of *ScriptOralia*, chapter Syntax gesprochener Sprachen, die Fragestellung, pages 1–18. Tübingen: Narr, 1990.

[PL92] Robert Pieraccini and Esther Levin. Stochastic representation of semantic structure for speech understanding. *Speech Communication*, 11:283–288, 1992.

[Pol90] Jordan B. Pollak. Recursive distributed representations. *Artificial Intelligence*, 46:77–105, 1990.

[Pol91] Jordan B. Pollak. The induction of dynamical recognizers. *Machine Learning*, 7:227–252, 1991.

[Pow89] David Powers. *Machine Learning of Natural Language*. Springer-Verlag, 1989.

[PR91] David Powers and Lerry Reeker, editors. *Machine Learning of Natural Language and Ontology*, volume D-91-09. DFKI, March 1991.

[PS93] Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. [acl93], pages 128–135.

[QR89] J. Ross Quinlan and Ronald L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models. IEEE, 1989.

[RB90] John Rager and George Berg. A connectionist model of motion and government in chomsky's government-binding theory. *Connection Science*, 2:35–52, 1990.

[Red] Fast Non-Linear Dimension Reduction. Nandakishore kambhatla and tod k. leen. neuroprose.

[Rei84] O. Reichmann. Historische lexikographie. *Sprachgeschichte*, 1. Halbband, 1984.

[Res92] Philip Resnik. A class based approach to lexical discovery. [acl92], pages 327–329.

[Rin92] Mark Ring. Learning sequential tasks by incremental adding higher orders. [nip92], pages 115–122.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, 1989.

[RK89] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[RL91] H. Ritter and E. Littmann. Generalization abilities of cascade network architectures. [nip91], pages 188–195.

[RMS+92] David Roe, Pedro J. Moreno, Richard W. Sproat, Fernando C.N. Rereira, Micheal D. Riley, and Alejandro Macarron. A spoken language translator for restricted-domain context free language. *Speech Communication*, 11:311–319, 1992.

[Roj93] R. Rojas. *Theorie der neuronalen Netze*. Springer Lehrbuch. Springer-Verlag, 1993.

[Roo93] Mats Rooth. Two-dimensional clusters in grammatical relations. 1993. draft.

[RS89] H. Ritter and K. Schulten. Convergence properties of kohonen's topology conserving maps: Fluctuations, stability, and dimension selection. *Biological Cybernetics*, 60:59–71, 1989.

[RSM90] Helge Ritter, Klaus Schulten, and Thomas Martinez. *Neuronale Netze: Eine Einführung in die Neuroinformatik selbstorganisierter Netzwerke*. Reiher Künstliche Intelligenz. Addison-Wesley, 1990.

[Sam69] John W. Sammon. A nonlinear mapping for data-structure analyis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[San88] Eugene Santos. A massivly parallel self-tuning context-free parser. [nip88], pages 537–544.

[Sch92] Hinrich Schütze. Word space. [nip92], pages 895–902.

[Sch93a] Jürgen Schmidhuber. *Netzwerkarchitekturen, Zielfunktionen und Kettenregel*. PhD thesis, Techniche Universität München, 1993. Habilitationsschrift.

[Sch93b] Hinrich Schütze. Exploring syntagmatic and paradagmatic relations in a large text corpus. 1993. to appear.

[Sch93c] Hinrich Schütze. Part of speech induction from scratch. 1993. to appear in ACL.

[Sch93d] Hinrich Schütze. Translation by confusion. 1993. to appear.

[Sch94] Georg Schifferdecker. Extraction of hierarchical structures in language. Diplomarbeit, Universität Karlsruhe (TH), 1994.

[Seg90] Jakob Segen. Graph clustering and model learning by data compression. In *Machine Learning, Proceedings of the Seventh International Conference*, 1990.

[Sen92] Stephanie Seneff. Tina: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86, 1992.

[Sha91] Noel E. Sharkey. Connectionists represenation techniques. *AI Review*, 5:153–167, 1991.

[SI92] Eiichiro Sumita and Hitoshi Iida. Example-based transfer of japanese adnominal particles into english. 1992.

[SL77] Detlef Steinhausen and Klaus Langer. *Clusteranalyse*. de Gruyter, 1977.

[Sma93] Frank Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993.

[Smo92] Paul Smolensky. Harmony grammar for formal languages. [nip92], pages 874–854.

[Sol64a] R. J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7:1–22, 1964.

[Sol64b] R. J. Solomonoff. A formal theory of inductive inference. part ii. *Information and Control*, 7:224–254, 1964.

[SSCM91] David Servan-Schreiber, Axel Cleeremans, and James L. McClelland. Graded state machines: The representation of temporal contigencies in simple recurrent networks. *Machine Learning*, 7:161–193, 1991.

[Sto91] Andreas Stolcke. Vector space grammars and the aquisition of syntactic categories. In Powers and Reeker [PR91], pages 174–179.

[Suh73] B. V. Suhotin. Methode de dechiffrage, outil de recherche en linguistique. *TA Informationes*, 2:3–43, 1973.

[SW94] B. Suhm and A. Waibel. Towards better language models for spontaneous speech. In *ICSLP*, Yokohama, Japan, 1994. submitted.

[UEGW93] Akiro Ushido, David A. Evans, Red Gibson, and Alex Waibel. Frequency estimation of verb subcategorization frames based on syntactic and multidimensional statistical analysis. 1993. appeared in ??

[UV92] K.P. Unnikrishnan and K.P. Venugopal. Learning in connectionists networks using the alopex algorithm. In *Proceedings of the IJCNN*, pages 1926–1931. IEEE Press, 1992.

[VHA92] Atro Voutilainen, Juha Heikkilä, and Arto Antilla. Constraint grammar of english: A performance-oriented introduction. Technical Report 21, University of Helsinki, Department of General Linguistics, 1992.

[WAWB+94] M. Woszyna, N. Aoki-Waibel, F. D. Buø, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, T. Schultz, B. Suhm, M. Tomita, and A. Waibel. Janus 93: Towards spontanous speech translation. In ICASSP-94 [ICA94].

[WB68] C.S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11:185–194, 1968.

[WBK92] Jerry B. Weinberg, Gautam Biswas, and Glenn R. Koller. Conceptual clustering with systematic missing values. In *Machine Learning, Proceedings of the Ninth International Conference*, 1992.

[Wei83] Ralph M. Weischedel. Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics*, 9(3-4):161–177, 1983.

[Wer89a] Stefan Wermter. Integration of semantic and syntactic constraints for structural noun phrase disambiguation. In *Eleventh International Joint Conference on Artificial Intelligence*, 1989.

[Wer89b] Stefan Wermter. Learning semantic relations in compound nouns with connectionists networks. In *Eleventh Annual Conference of the Cognitive Science Society*, 1989. Ann Arbor.

[Wer90] Stefan Wermter. Combining symbolic and connectionist techniques for coordniation in natural language processing. German Workshop on Artificial Intelligence (GWAI), Springer-Verlag, 1990. Eringerfeld.

[Wer93] Stefan Wermter. A hybrid connectionist architecture for a scanning understanding. [eca93], pages 188–192.

[WHH+89] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. Phoneme recognition using time-delay neural networks. IEEE, 1989.

[WL89] Stefan Wermter and Wendy G. Lehnert. A hybrid symbolic/connectionist model for noun phrase understanding. *Connection Science*, 1(3):255–271, 1989.

[Won92] E. K. Wong. Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3):287–303, 1992.

[WW94] Monika Woszczyna and Alex Waibel. Inferring linguistic structure in spoken language. In *Proceedings of the International Conference on Spoken Language Processing*. ASJ, 1994.

[XY92] Lei Xu and Alan Yuille. Self-organizing rules for robust principal component analysis. [nip92], pages 467–474.

[Yan93] Miin-Shen Yang. On a class of fuzzy classification maximum likelihood procedures. *Fuzzy Sets and Systems*, 57:365–375, 1993.

[Zel93a] Andreas Zell. Nessus handbuch. Technical Report 3, Universität Stuttgart, Institut für Parallele und verteilte Höchstleistungsrechner, 1993.

[Zel93b] Andreas Zell. Snns user manual, version 3.0. Technical Report 3, University of Stuttgart, Institute for Parallel and Distributed High Perfomance Systems, 1993.

[Zip35] G. Zipf. *The Psycho-Biology of Language*. Houghton Millin, 1935.