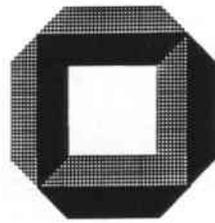


Diplomarbeit

**Einsatz Neuronaler Netzwerke
zur
Priorisierung von Elektronischer Post**

von

Petra Geutner



Institut für Logik, Komplexität und Deduktionssysteme
Fakultät für Informatik
Universität Karlsruhe
D – 7500 Karlsruhe, F.R.G.

Betreuer: Dipl.-Phys. U. Bodenhausen

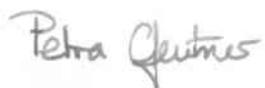
Erstgutachter: Prof. Dr. A. Waibel

September 1992

Erklärung

Ich erkläre hiermit, die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe angefertigt zu haben. Die verwendeten Literaturquellen sind im Literaturverzeichnis vollständig angegeben.

Karlsruhe, den 14. September 1992



Petra Geutner

Danksagung

Viele Personen haben mich direkt und indirekt bei der Anfertigung dieser Arbeit unterstützt. An dieser Stelle möchte ich ihnen deshalb meinen herzlichen Dank aussprechen.

Mein besonderer Dank gilt

Herrn Dipl.-Phys. U. Bodenhausen

Herrn Prof. Dr. A. Waibel

für die zuvorkommende Betreuung während meiner Tätigkeit.

Nicht zuletzt Dank an alle Institutsmitarbeiter und -angehörigen, die (fast) immer nur stillschweigend litten und geduldig all meine ressourcenverschlingenden Simulationen ertrugen.

Inhaltsverzeichnis

Inhaltsverzeichnis	i
Tabellenverzeichnis	v
Abbildungsverzeichnis	vii
1 Einleitung	1
2 Motivation und Aufgabenstellung	3
3 Einführung in Neuronale Netze	6
3.1 Entwicklung Neuronaler Netze	6
3.2 Neurophysiologische Grundlagen	7
3.3 Aufbau und Struktur Neuronaler Netze	9
3.4 Klassische Netzwerktypen	11
3.5 Lernen in Neuronalen Netzen	12
3.6 Einsatz Neuronaler Netze	13
4 Grundlagen	15
4.1 Natürliche Sprache	15
4.2 Adaptiver Erwerb von Sprache	17
4.3 Morphologie	21
4.4 Glättungstechniken	22
4.5 Backpropagation	23

5	ACEPS	30
5.1	Systemarchitektur	30
5.2	Funktionalitätsbeschreibung	31
5.3	Aufbau und Struktur des Netzwerkes	35
5.3.1	Entwurf eines Netzwerkes	35
5.3.2	Architektur des Netzwerkes	36
5.3.3	Der Trainingsalgorithmus	38
5.4	Realisierung	40
5.5	Anwendungen des ACEPS	41
6	Simulationen und Ergebnisse	43
6.1	Experimentelles Vorgehen	43
6.1.1	Vorgehen beim Training	45
6.1.2	Vorgehen beim Testen	45
6.1.3	Klassifikation durch Menschen	46
6.1.4	Beschränkung des Wortschatzes	55
6.2	Verwendete Verfahren	61
6.2.1	Einfache Klassifikation durch das ACEPS	61
6.2.2	Beschränkung der Eingabe	62
6.2.3	Realisierung von 3 Teilnetzen	65
6.2.4	Hinzunahme von Wortpaaren	67
6.2.5	Constructive Learning	69
6.2.6	Manipulation von Verbindungen	71
6.2.7	Reduktion auf lexikalische Wörter	72
6.2.8	Reduktion auf Morpheme	73
6.2.9	Rauschen	75
6.2.10	Glättung der Parameter (Smoothing)	76
6.2.11	Verwendung von Backpropagation als Trainingsalgorithmus	79
6.3	Zusammenfassung der Ergebnisse	81
6.4	Fehleranalyse	82
6.4.1	Performanz auf den Trainingsdaten	83
6.4.2	Performanz auf den Testdaten	85
6.4.3	Abschließende Betrachtungen	88

7 Zusammenfassung und Ausblick	89
Literaturverzeichnis	92
A Beispiel-Emails	96
A.1 Email in Originalform	96
A.2 Email nach Vorverarbeitung	98
A.3 Email nach Herausfiltern nicht deskriptiver Wörter	99
B Liste nicht deskriptiver Wörter (LNW)	100
C Wortlisten	101
C.1 Konferenzankündigungen	101
C.2 Fachliche Diskussion	102
C.3 Stellenangebote	102
C.4 Unwichtiges	103
C.5 Veröffentlichungen	104
D Beispiele falsch klassifizierter Emails	105
D.1 In „Konferenzankündigungen“ eingestuftes „Unwichtiges“	105
D.1.1 Beispiel 1	105
D.1.2 Beispiel 2	105
D.1.3 Beispiel 3	107
D.1.4 Beispiel 4	107
D.2 Fehlleistung der Vorklassifikation	108

Tabellenverzeichnis

4.1	XOR-Problem	24
6.1	Kenndaten der Test- und Trainingsmails	45
6.2	Menschliche Testergebnisse	47
6.3	Konfusionsmatrix Testperson 1	49
6.4	Konfusionsmatrix Testperson 2	49
6.5	Konfusionsmatrix Testperson 3	50
6.6	Konfusionsmatrix Testperson 4	50
6.7	Konfusionsmatrix aller Testpersonen im Durchschnitt	51
6.8	Konfusionsmatrix der Vorklassifikation	51
6.9	Übereinstimmung der Testpersonen mit der Vorklassifikation	52
6.10	Übereinstimmung der Testperson 4 mit der Vorklassifikation	52
6.11	Übereinstimmung der Testpersonen 1 und 2	53
6.12	Übereinstimmung der Testpersonen 3 und 4	54
6.13	Verteilung der Nachrichten pro Themengebiet	54
6.14	Ergebnisse einer einfachen Klassifikation durch das ACEPS	62
6.15	Simulationsergebnisse bei Beschränkung des Eingabetextes	63
6.16	Testergebnisse einer einfachen Klassifikation mit 3 Teilnetzen	66
6.17	Hinzunahme von Wortpaaren beim Training	69
6.18	Klassifikation mit <i>constructive learning</i>	70
6.19	Simulationsergebnisse der Experimente mit dem Wortschatz des ACEPS	74
6.20	Experimente mit dem Wortschatz des ACEPS und Wortpaaren	75
6.21	Simulationsergebnisse bei Anwendung von Rauschen	76
6.22	Simulationsergebnisse nach Einführung von <i>cooccurrence smoothing</i>	78

6.23	Ergebnisse der Simulationen mit Backpropagation durch das ACEPS	80
6.24	Zusammenfassung der Simulationsergebnisse	82
6.25	Konfusionsmatrix nach 1. Trainingszyklus	84
6.26	Konfusionsmatrix nach 11. Trainingszyklus	84
6.27	Konfusionsmatrix nach dem 1. Test	86
6.28	Konfusionsmatrix nach dem 11. Test	86
6.29	Performanz je Themengebiet nach 1. und 11. Test	87

Abbildungsverzeichnis

3.1	Nervenzelle des menschlichen Gehirns	8
3.2	Architektur eines neuronalen Netzwerkes mit n Schichten	9
3.3	Informationsfluß in einem Netzwerkelement j	10
4.1	Abbildung eines Satzes auf seine Bedeutung	18
4.2	Traditionelles überwachtes Lernen	19
4.3	Rückkopplung der Fehlermeldung auf semantischer Ebene	19
4.4	Netzwerk zur Repräsentation des XOR-Problems	24
5.1	Systemarchitektur des ACEPS	31
5.2	Funktionaler Ablauf während der Verarbeitung einer Email	33
5.3	Funktionaler Ablauf während der Lernphase des ACEPS	34
5.4	Das Netzwerk des ACEPS	36
5.5	Das Netzwerk des ACEPS nach Ergänzung durch eine Zwischenschicht	38
5.6	Systemarchitektur des ACEPS unter Einbeziehung des <i>Xmh</i> -Tools	42
6.1	Länge einer Nachricht je Klasse	55
6.2	Größe des Vokabulars einer Nachricht je Klasse	56
6.3	Länge einer Nachricht je Klasse nach Bearbeitung durch das ACEPS	56
6.4	Größe des Vokabulars einer Nachricht je Klasse nach Bearbeitung	57
6.5	Wachstum des Wortschatzes des ACEPS	57
6.6	Rang-Häufigkeits-Verteilung von Wörtern	58
6.7	Vergleich der Performanz verschiedener Eingabelängen	64
6.8	Vergleich der Trainingsleistung mit und ohne Hinzunahme von Wortpaaren	68
6.9	Vergleich der Trainingsergebnisse mit und ohne <i>constructive learning</i>	71
6.10	Vergleich der Testergebnisse mit und ohne <i>constructive learning</i>	72

6.11 Testperformanzmessungen des Experiments WP_ANF50	77
6.12 Ergebnisse unter Benutzung verschiedener Architekturen	83

Zusammenfassung

Traditionelle sprachverarbeitende Systeme basieren auf einem vordefinierten Vokabular und vorgegebenen Grammatik- bzw. Syntaxregeln. Natürliche Sprache, wie sie vom Menschen zur Kommunikation verwendet wird, läßt sich jedoch nur schwer einschränken. In dieser Arbeit wird ein System beschrieben, das sich mit der Verarbeitung von Nachrichten befaßt. Dabei wird der notwendige Sprachumfang durch ein neuronales Netzwerk im Laufe der Aufgabenstellung automatisch erworben. Grundlage des Netzwerkes ist ein adaptiver inkrementell lernender Algorithmus, der sich Wissen interaktiv aneignet und Nachrichten in Themengebiete bzw. Prioritätsstufen einordnet. Ohne die Notwendigkeit langer Trainingsläufe werden somit Assoziationen zwischen Nachrichten und sinnvollen Reaktionen auf diese Eingaben ermöglicht. In einer speziellen Anwendung wird das System als Informationsfilter bei der Verarbeitung von Nachrichten elektronischer Post eingesetzt und auf einer großen Datenbasis solcher in fünf subjektive Kategorien fallender Nachrichten evaluiert. Die Genauigkeit der besten aller hier vorgestellten konnektionistischen Architekturen entspricht der Leistung von Testpersonen, die die gleiche Aufgabe unabhängig durchführten.

Kapitel 1

Einleitung

Neuronale Netzwerke können von völlig verschiedenen Blickwinkeln aus betrachtet werden: Biologen und Neurophysiologen benutzen Modelle neuronaler Netzwerke vorwiegend zum Verständnis der biologischen Vorgänge im Gehirn. Psychologen untersuchen Lerntheorien anhand von Netzen. Physiker sehen vor allem physikalische Strukturen, die sich durch Energiefunktionen beschreiben lassen. Informatiker schließlich, und aus dieser Sicht heraus ist diese Arbeit entstanden, sehen neuronale Netzwerke als hochgradig parallel arbeitende, lernfähige und teilweise selbstorganisierende Informationsverarbeitungssysteme.

Neuronale Netzwerke und deren parallele Verarbeitungsweise sind eine Alternative zum herkömmlichen Computer-Paradigma, das sich auf sequentielle Verarbeitung von Programmanweisungen stützt. Sie sind in Aufbau und Konzeption stärker an der Funktionsweise des menschlichen Gehirns orientiert als an der Arbeitsweise der klassischen *von-Neumann-Architektur*. Das menschliche Gehirn ist robust und fehlertolerant. Eben dies ermöglichen auch künstliche neuronale Netzwerke durch ihren konnektionistischen Ansatz. Neuronale Netzwerke sind fähig, aus Beispielen zu lernen und anhand von Beispielen zu verallgemeinern bzw. zu abstrahieren. Neuronale Netzwerke können mit fuzzy, probabilistischer, gestörter und inkonsistenter Information umgehen und ermöglichen damit schnelles Erkennen und Vervollständigen komplizierter Muster sowie assoziatives Speichern und Abrufen von Informationen.

Gerade durch die Ähnlichkeit neuronaler Netzwerke mit dem Gehirn des Menschen und ihre Fähigkeit, die Arbeitsweise des menschlichen Gehirns innerhalb gewisser Grenzen nachzubilden, sind künstliche neuronale Netzwerke von großem Interesse. Insbesondere bei der Verarbeitung natürlicher Sprache sind noch viele Probleme ungelöst und die sogenannte „Künstliche Intelligenz“ ist der natürlichen noch weitaus unterlegen. Hier stellt der Versuch, menschliche sprachliche Verhaltensweisen und sprachliche Fähigkeiten mit Hilfe von Computern nachzubilden, noch immer eine große Herausforderung dar.

Die Leistung künstlicher sprachverarbeitender Systeme so zu verbessern, daß sie den Menschen von zeitaufwendigen und routinemäßigen Tätigkeiten befreien können, wird deshalb Gegenstand weiterer Forschung sein. Dabei stellen künstliche neuronale Netzwerke ein wichtiges Instrument zur Verfügung, um dieses Ziel zu erreichen. Diese am Institut für Logik, Komplexität und Deduktionssysteme der Universität Karlsruhe ent-

standene Diplomarbeit entwickelt ein textverarbeitendes System, das als Werkzeug zur Behandlung elektronischer Post praktisch einsetzbar ist.

Kapitel 2

Motivation und Aufgabenstellung

Im Zeichen der immer weiter fortschreitenden Technisierung der heutigen Welt bedienen sich immer mehr Menschen moderner Wege der Nachrichtenübermittlung. Dazu gehören Übertragungen durch Radio- und Fernsehgeräte, Telefon und Fax. Insbesondere zählt hierzu auch eine Kommunikationsform, die Ausgangspunkt für die weiteren Betrachtungen dieser Arbeit ist. Es handelt sich dabei um Nachrichten, die über Rechnernetze in die ganze Welt verschickt werden können und per Computer vom Menschen abrufbar sind: elektronische Post (Electronic Mail bzw. auch Email). Je mehr solche neuartigen Techniken der Kommunikation von Menschen angenommen und benutzt werden, desto mehr ergibt sich durch die entstehende Informationsflut ein Kommunikationskollaps, wie er auch schon bei konventionellen Kommunikationsmethoden zu beobachten ist.

Betrachtet man den herkömmlichen heute üblichen Briefverkehr, läßt sich folgendes feststellen: Eine immer größer werdende Papierflut erreicht täglich unsere Briefkästen. Neben wichtigen Geschäftsmitteilungen und privater Korrespondenz sind darunter auch mehr und mehr Werbe- und Reklamesendungen. Es wäre also manchmal bequem und zeitsparend, könnte diese Briefflut durch Vorverarbeitung etwas gefiltert werden. Ähnlich dem leitenden Angestellten eines Großunternehmens, dem ein guter Sekretär bzw. eine gute Sekretärin sehr viel Arbeit abnimmt, würden auch uns dann nur noch die wirklich interessanten Anschreiben erreichen. Sekretär bzw. Sekretärin sortieren eingegangene Post nach Wichtigkeit oder bestimmter Dringlichkeit und erledigen manche dieser Briefe, sofern sie in ihrem Kompetenzbereich liegen, selbständig.

Diese Vorgehensweise wäre auch bei elektronischer Post von Nutzen. Die immer größere Verbreitung und bessere Verfügbarkeit moderner Kommunikationsformen wie Fax und Email sorgt für ständig wachsende Informationsmengen. Je mehr Nachrichten uns per Rechner täglich erreichen, desto mehr Zeit verbringen wir mit Lesen und Beantworten teilweise uns oft unwichtig vorkommender „elektronischer“ Briefe. Ein dem Lesen einer Email vorgeschaltetes System, das als Informationsfilter dient, bedeutet eine Erleichterung für Benutzer einer *Internet*-Adresse und wäre daher eine große Hilfe im täglichen Umgang mit elektronischer Post. Für das Betriebssystem Ultrix existiert dafür bereits ein regelbasiertes Werkzeug namens *procm*ail, das anhand bestimmter Merkmale eine Email-Nachricht klassifiziert und sie entsprechend dieser Klassifikation bearbeitet. Die

Einordnung in Klassen bzw. Gruppen kann nach bestimmten Kriterien durchgeführt werden:

- die Mails werden je nach Namen des Absenders in verschiedene Gruppen eingeordnet
- eine Mail wird als zu einer bestimmten Gruppe gehörig erkannt, falls ein bestimmtes Schlüsselwort in der „subject“-Zeile¹ vorkommt

Ein solches Vorgehen weist zwei entscheidende Nachteile auf. Zum einen erfordert es die zeitaufwendige Definition spezieller Schlüsselwörter, nach denen klassifiziert werden soll. Zum anderen geht der Empfänger eines durch elektronische Post an ihn geschickten Briefes anders vor. Er wird sich nicht damit zufriedengeben, nur den Absender bzw. die „subject“-Zeile der Nachricht zu lesen und den Brief dementsprechend zu klassifizieren. Vielmehr wird er auch den gesamten Wortlaut der Mail studieren, ihn zumindest grob überfliegen, und erst dann eine Entscheidung treffen.

In der vorliegenden Arbeit werden Möglichkeiten aufgezeigt, eine solche Art der Vorverarbeitung elektronischer Post mit Hilfe eines textverarbeitenden Systems durchzuführen. Die Kriterien, nach denen ankommende Nachrichten bearbeitet werden, werden vom jeweiligen Benutzer des Systems bestimmt. Dabei kann es sich um eine Einordnung nach Wichtigkeit der Email für den Empfänger handeln oder auch um die Festlegung einer Dringlichkeitsstufe, die die sofortige oder spätere Beantwortung der jeweiligen Mail betrifft. Genauso gut können aber auch verschiedene Themenschwerpunkte vom Benutzer definiert werden und es findet eine Klassifikation in Fachgebiete statt. Das im Rahmen dieser Arbeit entwickelte System, das ACEPS (A d a p t i v e C o n n e c t i o n i s t E m a i l P r o c e s s i n g), berücksichtigt nicht nur Absender und Gegenstand einer elektronischen Postnachricht, sondern geht auch auf den Inhalt einer Email ein. Aufgrund des Textkorpus wird die Email untersucht und durch ein neuronales Netzwerk in einen vom Benutzer des Systems vorgegebenen Rahmen eingepaßt.

Das neuronale Netzwerk verfügt nicht über ein vordefiniertes Vokabular und eine vorgegebene Syntax. Im Gegensatz zu vielen herkömmlichen Systemen, die natürliche Sprache verarbeiten, erwirbt es sich selbst automatisch ein Sprachmodell. Das Netz kennt die Konzepte *Wort* und *Phrase*, hat aber keine Instantiierungen dieser Konzepte. Mit Hilfe semantischer Information baut es sich ein Modell für die Sprache einer Aufgabenstellung auf. Die Gewichte der Verbindungen des Netzwerkes werden durch einen Trainingsalgorithmus bestimmt, der die Vorteile adaptiven und schnellen Lernens mit denen der generalisierten Delta-Regel verknüpft². Durch seinen adaptiven Trainingsalgorithmus benötigt das ACEPS keine vordefinierten Konzepte, sondern lernt während seiner Benutzung interaktiv und inkrementell. Diese Art des Lernens ermöglicht eine flexible Anpassung an sich ändernde Benutzerwünsche. Das Netzwerk empfängt als Eingabe Nachrichten der elektronischen Post und versucht, diesen geschriebenen Text in die vom Benutzer vordefinierten Dringlichkeitsstufen oder Themenschwerpunkte einzuordnen. Verknüpft mit dieser Einordnung sind semantische Aktionen, die je nach Klassifikation der Email ausgeführt werden. Dabei kann es sich um eine automatische Beantwortung der Mail handeln,

¹Stichworte, die den Gegenstand der Nachricht zusammenfassen.

²Näheres zum Trainingsalgorithmus des ACEPS in Kapitel 5.

das Weiterleiten einer Mail an einen anderen Empfänger oder auch um ein automatisches Löschen einer Nachricht, die als völlig unwichtig und nicht lesenswert eingestuft wurde. Damit fungiert das textverarbeitende System quasi als der bzw. die oben beschriebene „Sekretär“ oder „Sekretärin“, die die ankommenden Nachrichten vorverarbeitet und dem Empfänger das Lesen und Beantworten einer großen Masse von Post erleichtert.

Kapitel 3

Einführung in Neuronale Netze

Das folgende Kapitel stellt eine Einführung in die Grundbegriffe natürlicher und künstlicher neuronaler Netzwerke dar. Zunächst wird die Entwicklung künstlicher neuronaler Netze in den letzten 50 Jahren bis zur heutigen Zeit grob skizziert und die innerhalb dieser Zeit aufgetretenen Probleme und deren Lösungen geschildert. Es folgt eine kurze Einführung in die Strukturen des menschlichen Gehirns. Die dort vorhandenen Neuronennetze sind die Grundlagen für die hier vorgestellte allgemeine Architektur eines neuronalen Netzwerkes. Aufbauend auf dieser allgemeinen Struktur sind verschiedene klassische Netzwerktypen entstanden, die sich unterschiedlicher Lernmethoden bedienen. Ein Überblick über verschiedene Lernstrategien neuronaler Netzwerke und eine Diskussion der möglichen Anwendungsgebiete dieser Netzwerke schließen dieses Kapitel ab.

3.1 Entwicklung Neuronaler Netze

Konnektionistische Ideen in der Psychologie nehmen ihren Anfang schon bei Aristoteles. In unserem Jahrhundert [And88] beginnt die Erforschung neuronaler Netzwerke bereits 1943 mit einer Arbeit von McCulloch und Pitts über die Logik von Netzwerken mit binären Schwellwertelementen, sogenannten *McCulloch-Pitts-Neuronen* [McC43]. Fünfzehn Jahre später, 1958, entwickelte Rosenblatt das *Perceptron* [Ros58], ein sich teilweise selbstorganisierendes Netzwerk, das lernfähig ist und Erfahrungen verallgemeinern kann. Für die einfachste Klasse des Perceptrons mit nur einer Schicht konnte Rosenblatt die Konvergenz seines Lernalgorithmus nachweisen. 1969 kam es zu einem Rückschlag in der künstlichen neuronalen Netzwerkforschung als Minsky und Papert [Min88] zeigten, daß ein Perceptron in der Regel nicht in der Lage ist, einfache logische Funktionen wie z.B. das Exklusive-Oder (XOR) Problem¹ zu lösen. Damit ruhte die Forschung fast 20 Jahre lang und nur wenige Wissenschaftler beschäftigten sich weiter mit neuronalen Netztheorien. 1982 veröffentlichte Hopfield die Ergebnisse seiner Arbeit: das *Hopfield-Modell* [Hop82], in dem er Methoden und Techniken der theoretischen Physik durch Einführung einer Energiefunktion mit denen neuronaler Netze verknüpfte. Damit nahm das Interesse an neuronalen Netzwerken wieder zu. Die Schwächen des Perceptrons beruhen vor allem auf

¹Für eine nähere Erklärung des XOR Problems siehe Abschnitt 4.5.

der Tatsache, daß dieses Modell nur eine Schicht lernfähiger Netzwerkelemente enthält. Für mehrlagige Netzwerke existierte noch immer keine Möglichkeit, die Gewichte der verdeckten Schichten geeignet anzupassen. Rumelhart, Hinton und Williams entwickelten 1986 schließlich *Backpropagation* [Rum88], einen Lernalgorithmus, der auch die Berechnung von Fehlersignalen für die Neuronen verdeckter Schichten erlaubt. Wenn auch keine universellen Problemlöser, so sind Backpropagation-Netzwerke doch äußerst vielseitige Netzwerkmodelle. Neuronale Netzwerke werden in einer Vielfalt verschiedener Anwendungsbeispiele eingesetzt, die jedoch eines gemeinsam haben: sie lassen sich fast alle auf mehr oder weniger komplexe Muster- und Gestaltungsprobleme zurückführen. Anwendungen dieser Art sind in der Regel schneller und einfacher mit künstlichen neuronalen Netzwerken als mit anderen Methoden zu realisieren.

Nicht zuletzt aus diesem Grund erleben neuronale Netzwerke seit Anfang der 80er Jahre wieder einen Aufschwung. Zur Zeit der Erstellung dieser Diplomarbeit (1992) läßt sich sogar von einem richtiggehenden „konnektionistischen Boom“ sprechen, wobei Backpropagation-Netzwerke den am häufigsten eingesetzten Netzwerktyp darstellen.

Eine Vielzahl von Literatur befaßt sich mit neuronalen Netzen und nicht weniger Forschungsprojekte bedienen sich konnektionistischer Techniken. Viele dieser Projekte werden noch einige Hindernisse überwinden müssen, um ihr — oft hochgestecktes und ehrgeiziges — Ziel zu erreichen: die Arbeitsweise des menschlichen Gehirns teilweise nachzubilden und damit Menschen unangenehme oder Routinearbeiten zu erleichtern. Demgegenüber gibt es jedoch eine ganze Reihe praktischer Anwendungen (siehe Abschnitt 3.6), ausgehend von Gestalt- und Handschriftenerkennung über die Robotik bis hin zur Prognose von Aktienkursen, die auch schon kommerziell vertrieben und genutzt werden. Marktforscher prognostizieren der konnektionistischen Technik bis in die 90er Jahre jährlich zweistellige Wachstumsraten. Die Zukunftsaussichten sind also hervorragend.

3.2 Neurophysiologische Grundlagen

Wie schon in Kapitel 1 erwähnt, orientieren sich neuronale Netzwerkarchitekturen an Modellen des menschlichen Gehirns und dessen Nervenzellen. Ohne sich allzusehr in Details zu verlieren, ist nachfolgender Abschnitt eine kurze Einführung in die Strukturen des Gehirns.

Das menschliche Gehirn besteht aus insgesamt etwa 10-100 Milliarden Nervenzellen bzw. Neuronen. Sie sind die morphologisch-anatomischen und funktionellen „Grundeinheiten“ im Nervensystem. Die genaue Arbeitsweise des Gehirns ist auch heute noch Gegenstand der Forschung. Während die größeren Zusammenhänge bisher weitgehend unbekannt sind, gibt es bereits detailliertere Kenntnisse über die einzelnen Neuronen [Day90]. Sie haben die Aufgabe der Signalübertragung und -verarbeitung. Dies erfolgt mit Hilfe elektrischer und biochemischer Vorgänge. Bei einer durchschnittlichen *Konnektivität* (= Anzahl ausgehender bzw. ankommender Verbindungen) von etwa 10.000 ergibt sich bei 10-100 Milliarden Neuronen eine Zahl von 100-1.000 Billionen (also 10^{14} - 10^{15}) möglicher Verbindungen. Durch diese Verbindungen werden die Nervensignale zwischen den einzelnen

Neuronen weitergeleitet. Lernen erfolgt im wesentlichen durch die Modifikation der Verbindungen. Abbildung 3.1 aus [Her91] zeigt eine typische Nervenzelle des menschlichen Gehirns.

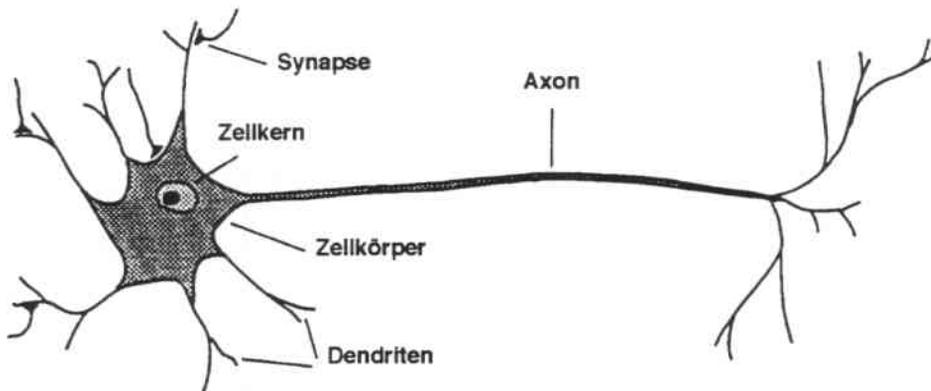


Abbildung 3.1: Nervenzelle des menschlichen Gehirns

Der Eingabebereich der Zelle besteht aus einer Menge von Fasern, die *Dendriten* genannt werden. Der Ausgabebereich eines Neurons ist eine lange, verästelte Faser, die als *Axon* bezeichnet wird. Durch die Nervenzellen wird ein Impuls angeregt, der entlang der Axon-Äste bis an die Enden dieser Faser gesendet wird. Der Verbindungspunkt zwischen einem Axon und einem Dendrit heißt *Synapse*. Wenn eine Folge von Impulsen am dendritischen Bereich eines Neurons empfangen wird, erhält man als Ergebnis üblicherweise eine erhöhte Wahrscheinlichkeit, daß das Zielneuron einen Impuls zu seinem Axon senden wird.

Die Verarbeitungsgeschwindigkeit der Neuronen ist im Vergleich mit der Leistung heutiger Computer um den Faktor 10^6 langsamer [Her91]. Ihre Anzahl sowie die Menge ihrer Verknüpfungen ist dagegen unvorstellbar groß und für die heutige Computertechnik unerreichbar. Leistung und Schnelligkeit des Gehirns beruhen also vor allem auf hoher Parallelität, flexibler Lernfähigkeit und der Tatsache, auch bei teilweise massiven Schädigungen noch funktionsfähig zu bleiben. Diese Eigenschaften machen eine Nachahmung der Architektur des menschlichen Gehirns erstrebenswert. Ohne zuviel Wert auf die genaue Nachbildung der biologischen Strukturen zu legen, versuchen künstliche neuronale Netzwerke die Vorzüge des Gehirns mit der Leistungsfähigkeit heutiger Computer zu verbinden.

3.3 Aufbau und Struktur Neuronaler Netze

Ein künstliches neuronales Netzwerk besteht in der Regel aus mehreren Schichten. Dabei handelt es sich mindestens um Eingabe- und Ausgabeschicht². Dazwischen kann eine beliebige Anzahl von verdeckten Schichten liegen. Ihre Anzahl hängt von Art und Eigenschaften des zu lösenden Problems ab. In der Regel verläuft der Informationsfluß zwischen den einzelnen Schichten von der Eingabe- in Richtung der Ausgabeschicht. Jede Schicht wiederum besteht aus einzelnen Basiselementen, den sogenannten Neuronen oder *Elementen*. Elemente einer Schicht sind durch *Verbindungen* mit Elementen derselben Schicht oder Teilen anderer Schichten verknüpft. Die Verbindung eines Elements i zu einem Element j wird als Gewicht w_{ij} bezeichnet. Bild 3.2 stellt ein neuronales Netzwerk mit n Schichten dar.

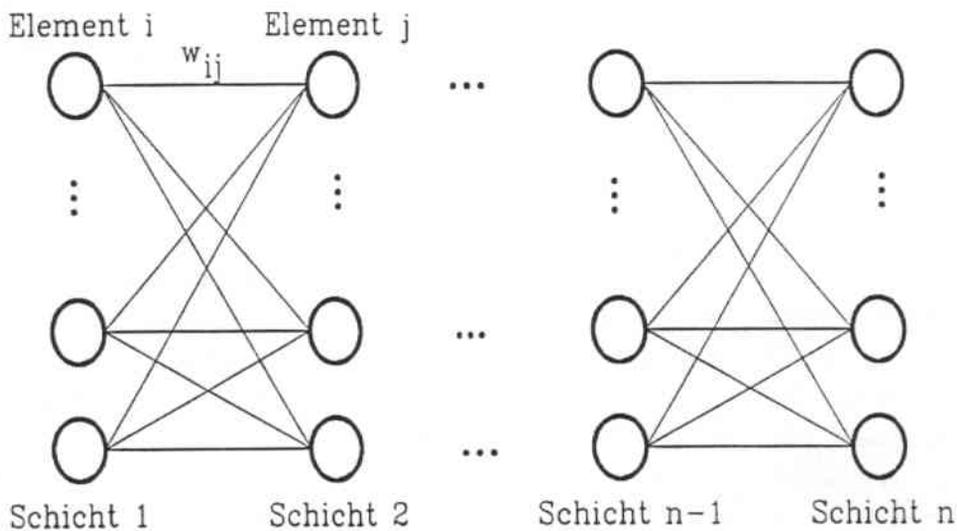


Abbildung 3.2: Architektur eines neuronalen Netzwerkes mit n Schichten

Viele Netzwerke bestehen außerdem noch aus einem Element, das keiner der vorher genannten Schichten zuzuordnen ist: dem sogenannten *Bias*. Der Bias hat den konstanten Wert $+1$ und dient als eine Art Schwellwertfunktion.

Für jedes Element eines Netzwerkes sind drei verschiedene Funktionen definiert [Was89]:

- Inputfunktion
- Aktivierungsfunktion
- Outputfunktion

Die Inputfunktion bereitet die in ein Element führenden Eingänge auf. Das kann z.B. in einer einfachen Summation aller Eingänge o_i multipliziert mit dem jeweiligen Gewicht der Verbindung von Element i nach j bestehen:

²Auf Architekturen wie das Hopfield-Modell oder die Boltzmann-Maschine, die von diesem Aufbau abweichen, wird in diesem Abschnitt nicht weiter eingegangen.

$$net_j = \sum_i o_i \cdot w_{ij}.$$

Das Ergebnis der Inputfunktion dient innerhalb des Netzelements als Eingabe für die Aktivierungsfunktion. Durch diese Funktion wird die Aktivität des Elements festgelegt, die sie in Abhängigkeit von der Eingabe haben soll. Die einfachste Aktivierungsfunktion ist

$$F(net_j) = 1 \cdot net_j.$$

Diese Aktivierungsfunktion ist *linear*, d.h. die Aktivität des Elements j entspricht dem Ergebnis der Inputfunktion multipliziert mit einem konstanten Faktor, in diesem Fall der Zahl 1. Eine andere häufig benutzte Aktivierungsfunktion ist auch die Sigmoid-Funktion, die in Abschnitt 4.5 näher erläutert wird. Weitere übliche Aktivierungsfunktionen sind der Tangens Hyperbolicus, die Sinus-Funktion, die Signum-Funktion oder auch die einfache Step-Funktion. Auf die Aktivierungsfunktion folgt die Outputfunktion. Die einfachste und meist benutzte Outputfunktion ist die direkte Abbildung der Aktivierung a_j eines Netzelements auf die Ausgabe o_j dieses Elements:

$$o_j = a_j.$$

Da es sich dabei um Identität von Aktivierung und Ausgabe handelt, findet in vielen Netzwerken keine explizite Definition einer Outputfunktion statt. Andere Outputfunktionen ermöglichen einen Wettbewerb zwischen den einzelnen Elementen einer Schicht. Bei einer *winner take all* Strategie wie sie bei konkurrierendem Lernen (*competitive learning*) [Gro88, Day90, Hec91] verwendet wird, würde z.B. nur das Element seinen Wert weitergeben, dessen Aktivierungsfunktion den höchsten Wert liefert, alle anderen haben eine Ausgabe gleich Null.

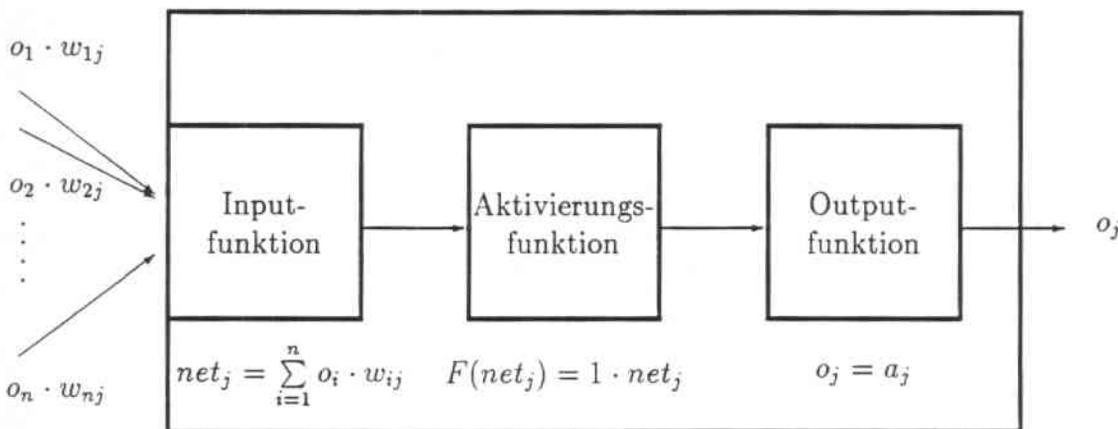


Abbildung 3.3: Informationsfluß in einem Netzelement j

Bild 3.3 zeigt den Informationsfluß innerhalb eines Elements j eines Netzwerkes. Die Inputfunktion besteht aus einer einfachen Summation aller Eingänge, die Aktivierungsfunktion ist linear, die Outputfunktion bildet die Aktivität des Elements j direkt auf die Ausgabe ab.

3.4 Klassische Netzwerktypen

Neuronale Netzwerke lassen sich in verschiedene Kategorien einteilen. Sie können aus einer oder mehreren Schichten bestehen, der Informationsfluß kann in nur eine Richtung oder mit Rückkopplung stattfinden, Lernen kann überwacht erfolgen oder das Netzwerk kann sich selbst organisieren (siehe 3.5). Das klassische Netzwerkmodell ist das von Rosenblatt 1958 vorgestellte *Perceptron* [Ros58], ein schichtenorientiertes neuronales Netzwerk, das auf nicht triviale Problemstellungen anwendbar ist. Heute bezeichnet „Perceptron“ ein rückkopplungsfreies (*feed-forward*) Netzwerk ohne verdeckte Schichten. Einfache binäre Schwellwertelemente stellen die Elemente dar. Das Perceptron lernt durch vorklassifizierte Ein-/Ausgabepaare, es handelt sich dabei um eine Form des überwachten Lernens. Für die im Perceptron benutzte Lernregel läßt sich das sogenannte Perceptron-Konvergenz-Theorem beweisen: jede Aufgabenstellung, die mit Hilfe eines Perceptrons darstellbar ist, kann in einer endlichen Anzahl von Schritten gelernt werden. Das Perceptron ist jedoch nicht für alle Problemklassen geeignet. So ist z.B. die Exklusive-Oder (XOR) Funktion nicht durch ein Perceptron lösbar, da es sich um ein nicht linear trennbares Problem handelt.

Dem Perceptron ähnlich ist das 1956-1960 von Widrow und Hoff entwickelte *Adaline-Modell* (*adaptive linear neuron*) [Wid60]. Ein Adaline-Netzwerk besteht aus einem einschichtigen rückkopplungsfreien Netz mit ebenfalls binären Schwellwertelementen. Durch eine neue Lernregel, die *Delta-Regel*, wird schnelleres Trainieren als beim Perceptron möglich. Die Delta-Regel setzt, wie schon beim Perceptron, ebenfalls überwacht Lernen voraus. Mit Hilfe eines speziellen Fehlersignals werden die Gewichte neu angepaßt, das Netzwerk „lernt“ dadurch schneller.

Das Problem rückkopplungsfreier Netzwerke besteht vor allem darin, die Gewichte der von außen nicht zugänglichen verdeckten Schichten in geeigneter Weise anzupassen. Eine dahingehende Erweiterung der im Adaline-Netzwerk verwendeten Delta-Regel findet sich im 1986 von Rumelhart, Hinton und Williams [Rum88] vorgestellten *Backpropagation-Algorithmus*: die *generalisierte Delta-Regel*. Diese Lernregel erlaubt die Berechnung eines Fehlersignals auch für die verdeckten Schichten eines Netzwerkes und ermöglicht somit das Lernen interner Repräsentationen. Mit Hilfe dieser Repräsentationen sind auch nicht linear trennbare Probleme durch neuronale Netzwerke lösbar (näheres dazu siehe auch Abschnitt 4.5).

Bei den bisher vorgestellten Netzwerktypen handelt es sich um rückkopplungsfreie Netzwerke. Dieser Netzwerktyp liefert eine definierte, eindeutige Antwort auf jede Eingabe, d.h. eine konstante Eingabe produziert eine konstante Ausgabe. Im Gegensatz dazu steht das 1982 von Hopfield entwickelte *Hopfield-Netz* [Hop82]. In diesem Netz wird die Ausgabe des Systems wieder als Eingabe verwendet und das Verhalten des Netzwerkes ändert sich eventuell mit jedem Durchlauf. Eine eindeutige Antwort liegt erst dann vor, wenn das System nach endlich vielen Durchläufen einen stabilen Zustand erreicht. Dies ist der Fall, wenn eine gerade erzeugte Ausgabe, die als Eingabe des Netzwerkes verwendet wird, sich selbst reproduziert.

Beim Hopfield-Modell handelt es sich um ein einschichtiges Netzwerk, das vollständig und symmetrisch vernetzt ist. Die verwendete Lernregel ist eine Erweiterung der Hebbischen

Lernregel [Heb49] (siehe auch im nächsten Abschnitt) und wird als generalisierte Hebb-Regel bezeichnet. Wie schon bei den vorher erwähnten Netzwerktypen findet auch beim Hopfield-Modell überwachtes Lernen statt. Der nächste Abschnitt geht detaillierter auf die verschiedenen Lernstrategien bei neuronalen Netzen ein.

3.5 Lernen in Neuronalen Netzen

Eine der wichtigsten Fragen bei der Verwendung eines neuronalen Netzwerkes ist das Problem der Programmierung bzw. des Lernens des Netzes: Wie müssen die Gewichte der Verbindungen zwischen den einzelnen Elementen gewählt werden, damit das Netzwerk eine spezielle Aufgabe erfüllen kann?

Nur sehr wenige Probleme bieten im voraus genügend Wissen über das Netzwerk, um bereits bei seinem Aufbau die Gewichte a priori einzustellen. In fast allen Fällen muß dem Netzwerk beigebracht werden, welche Berechnung ausgeführt werden soll. Dies geschieht durch iterative Anpassung der Stärke der Gewichte während der Lernphase. Das Lernen kann dabei auf zwei verschiedene Methoden [Was89] durchgeführt werden:

Überwachtes Lernen: Beim überwachten Lernen wird die Ausgabe des Netzwerkes mit der gewünschten korrekten Ausgabe verglichen und aus eventuell gemachten Fehlern gelernt. Unter überwachtes Lernen oder Lernen mit einem Lehrer fällt auch Verstärkungslernen (*reinforcement learning*), wo die einzige Rückmeldung eine Anerkennung oder ein Abweisen der Ausgabe des Netzwerkes ist. Dabei wird beim Abweisen einer inkorrekten Ausgabe keine Korrektur des Fehlers vorgenommen.

Unüberwachtes Lernen: Ist das Lernziel eines Netzes nicht genau definiert (z.B. in Form von Beispielen mit korrekten Ein-/Ausgabepaaren), muß es unüberwacht lernen. Die einzige Informationsbasis für den Lernvorgang des Netzwerkes sind dann die Beziehungen der Eingabemuster untereinander. Das Netzwerk bildet aus diesen Beziehungen selbst Kategorien und erzeugt Ausgabesignale, die der Kategorie des Eingabemusters entsprechen.

Die Auswahl einer geeigneten Lernregel beeinflusst die Leistungsfähigkeit eines Netzwerkes. Bereits 1949 veröffentlichte Hebb seine biologisch motivierte Hypothese [Heb49], nach der die Verbindung zwischen zwei Neuronen proportional zur Stärke der Korrelation zwischen den Aktivitäten beider Neuronen verändert wird. Von der Hebb'schen Lernregel angefangen, wurde immer weiter nach geeigneten Lernalgorithmen für neuronale Netzwerke geforscht, was schließlich zur generalisierten Delta-Regel des Backpropagation-Netzwerkes führte (siehe auch Abschnitt 4.5). Mit Hilfe dieser Lernregel ist ein neuronales Netz in der Lage, interne Repräsentationen zu erlernen und damit auch nicht linear trennbare Probleme zu lösen, die ohne verdeckte Schichten nicht zu bewältigen sind [Min88].

3.6 Einsatz Neuronaler Netze

Neuronale Netzwerke können aus Erfahrung lernen und müssen deshalb nicht wie konventionelle Computer programmiert, sondern lediglich trainiert werden. Viele praktische Anwendungen verfügen über große Datenmengen, auf denen operiert werden muß. Implizite Regelmäßigkeiten in diesen Datenmengen müssen selbständig erkannt werden. Ebenso ist eine schnelle Anpassung an veränderte Bedingungen erforderlich. Durch die Art und Weise ihres Lernvorgangs können neuronale Netzwerke diese Anforderungen erfüllen. Dadurch verfügen sie über Eigenschaften, die sie für Anwendungen in bestimmten Bereichen besonders vorteilhaft machen [Hec91, Schö90]. So sind in den USA Anwendungen im Banken- und Finanzbereich bereits an der Tagesordnung. Bonitätsanalysen von Privatkunden, Untersuchungen von Risikomustern und -klassen finanzieller Engagements sowie Prognosen von Markttendenzen und Aktienkursen sind nur einige Beispiele dafür.

Ein weiteres klassisches Anwendungsgebiet neuronaler Netzwerke ist die Verarbeitung von natürlicher Sprache. Dazu zählen sowohl Sprachgenerierung als auch Sprachanalyse bzw. Sprachverstehen. Es existieren bereits Systeme³, die geschriebenen Text in akustische Sprachsignale umsetzen. Dabei ist die Qualität der Aussprache, mit der der vorgegebene Text laut vorgelesen wird, vergleichbar mit der Aussprache eines 5-6jährigen Kindes. Rumelhart entwickelte ein Modell, das in der Lage ist, einfache Sprachregeln selbständig zu erlernen. So lernt sein System z.B. implizite Regeln zur Vergangenheitsform von englischen Verben, ohne diese zuvor in irgendeiner Form gekannt zu haben, was einer Form der Generalisierung entspricht.

Handschriften maschinell lesen und auswerten zu können, ist ein weiteres mögliches Einsatzgebiet neuronaler Netzwerke. Aufgrund großer Abweichungen zwischen verschiedenen Handschriften ist dieses Problem mit konventionellen Methoden nicht zu lösen. Die Auswertung von Formularen, die handschriftlich ausgefüllt wurden, sind gerade bei Banken oder Behörden von Bedeutung. Auch die automatische Eingabe eines handgeschriebenen Textes über Scanner anstelle der Tastatur würde die Arbeit eines Menschen in vielen Bereichen erleichtern⁴. Cooper hat ein Netzwerkmodell entwickelt, das eine Klassifikation beliebiger geometrischer Figuren, also auch handgeschriebener Texte, mit hoher Genauigkeit erlaubt. Nestor, die Firma Coopers, hat auf dieser Basis ein Handschriftenerkennungssystem entwickelt. Die Benutzer des Systems können durch ein an einen Computer angeschlossenes Schreibgerät handschriftliche Zeichen ohne Tastatur eingeben.

Bei der Übertragung von Bildern oder Sprachsignalen über Leitungen mit geringer Bandbreite, die in Realzeit stattfinden soll, ergibt sich das Problem der Datenkompression. Die Übertragung kann durch Komprimierung der Daten beschleunigt werden, und es wird dafür gesorgt, daß keine relevanten Informationen verlorengehen. Cottrell, Munro und Zipser haben ein selbstorganisierendes dreilagiges Netzwerk entwickelt [Cot87], das digitalisierte Bilder in ihren verschiedenen Graustufen um den Faktor 8 komprimiert und nach der Übertragung mit hoher Detailtreue wieder regeneriert.

³Alle folgenden Anwendungsbeispiele in diesem Kapitel sind, sofern nicht anders angegeben, dem Buch [Schö90] entnommen.

⁴Die Autorin denkt dabei an die handschriftlichen Entwürfe dieser Arbeit.....

Komplexe Optimierungsprobleme, wie sie im Bereich des Operations Research auftauchen, sind aufgrund kombinatorischer Abhängigkeiten oft nur mit exponentiellem Rechenaufwand lösbar. Hopfield zeigte, daß solche kombinatorischen Probleme durch das von ihm entwickelte Netzwerkmodell effizient gelöst werden können. Mit Hilfe einer geeigneten Energiefunktion ist das Netzwerk in der Lage, Lösungen für Optimierungsprobleme als stabile Zustände zu speichern.

In Produktionsbetrieben der Industrie werden zur Sicherung der Qualität häufig automatisierte Stichproben entnommen. In der Automobilindustrie werden z.B. Motoren anhand ihrer Laufgeräusche auf Produktionsfehler überprüft. Dazu hat die Firma Siemens Netzwerke entwickelt, die defekte Motoren erkennen — eine Diagnose, zu der sonst nur erfahrene Kfz-Mechaniker in der Lage sind.

Für die Zukunft ist die Prozeßsteuerung und Regelungstechnik ein potentielles Einsatzgebiet neuronaler Netzwerke. Komplizierte Feinabstimmung von Bewegungsabläufen bei Montage- und Industrierobotern oder autonomen Systemen und die Steuerung von Einschwingvorgängen in der Regelungstechnik können durch neuronale Netze vorgenommen werden. Es ist somit zu erwarten, daß konventionelle Regelsysteme dann durch künstliche neuronale Netze ersetzt werden können.

Kapitel 4

Grundlagen

In Kapitel 2 wurde bereits die Idee für ein textverarbeitendes System diskutiert. In diesem Kapitel werden die Grundlagen beschrieben, auf denen das im Rahmen dieser Arbeit entwickelte System aufbaut. Dazu gehört eine kurze Einführung in natürliche Sprache und die Möglichkeit eines adaptiven Erwerbs von Sprache, der mit Hilfe eines speziellen Algorithmus vorgenommen wird. Um Wörter der englischen Sprache auf ihre Grundbedeutung zu reduzieren, wird kurz die Theorie und Methodologie eines Systems vorgestellt, das grammatikalische Formen englischer Wörter analysiert, und damit die Verwendung der kleinsten bedeutungstragenden Einheit eines Wortes ermöglicht. Durch die uneingeschränkte Eingabe von Text weist das Netzwerk des ACEPS eine riesige Anzahl von Parametern auf, der nur eine kleine Menge von Trainingsmustern gegenübersteht. Zur Glättung der geschätzten Wahrscheinlichkeiten wird eine eigens dafür entwickelte Form des *Smoothing* vorgestellt. Als weitere theoretische Grundlage des im Rahmen dieser Arbeit entstandenen Systems wird im letzten Abschnitt schließlich in die Vorgehensweise beim Lernen innerhalb des *Backpropagation*-Algorithmus eingeführt.

4.1 Natürliche Sprache

Jegliche Versuche, einige Facetten menschlicher Sprachbeherrschung auf den Computer zu übertragen, sind bisher nur unvollständig geglückt. Erst seit der Computer als Werkzeug der Sprachverarbeitung benutzt wird, stellt sich die Frage, mit welchen Datenmengen der menschliche Geist hantiert, wenn er Sprache gebraucht. Ein weiteres Problem ist, in wie weit sich eine natürliche Sprache überhaupt formalisieren läßt. Laut [Zim90] strotzt die heutige deutsche Orthographie nur so von Willkürlichkeiten und ist daher gegen jeden Algorithmus immun. Die englische Sprache, die die Grundlage dieser Arbeit bildet, verfügt zwar nicht über die gleiche Vielzahl von Grammatikregeln und Sonderfällen wie sie im Deutschen auftreten, jedoch ist auch ihre Formalisierung schwierig. Bei den Versuchen, dem Computer etwas Sprachvermögen beizubringen, haben Computer deutlich gemacht, eine wie überaus komplexe Leistung das Gehirn auf jeder Ebene der Sprachverarbeitung vollbringt. Trotz dieser fast hoffnungslosen Ausgangslage ist es gelungen, Computern sprachliche Fertigkeiten zu vermitteln, die sie schon zu nützlichen Werkzeugen

der Sprachverarbeitung machen. Dennoch wird weiter versucht, menschliche Sprach- und Verständnisfähigkeiten auf Computer zu übertragen.

Das Hauptproblem dieser Arbeit ist zunächst weniger die grammatikalische Vielfaltigkeit, die Inkonsequenzen und Ambiguitäten von Sprache, sondern eher der große Wortschatz eines Menschen. Die Sprachwissenschaft schätzt den deutschen Wortschatz [Zim90] auf etwa 400.000 Wörter. Der Gesamtbestand deutscher Allgemeinwörter in allen Fach- und Sondersprachen wurde 1986 auf über 30 Millionen geschätzt. Diese Millionen von Wörtern werden aus den sogenannten Morphemen¹ und Beugungsendungen gebildet, wobei die Zahl der Morpheme nicht mehr als 5.000 beträgt.

Im Vergleich zur deutschen Sprache wird der allgemeine englische Wortschatz laut [Zim90] sogar auf 700.000 Wörter geschätzt. Dafür haben englische Wörter eine beinahe unveränderliche Gestalt: Auf jedes Lexem² entfallen im Durchschnitt nur etwa zwei Flexeme³. In der deutschen Sprache existieren so viele Deklinations- und Konjugationsformen, daß durchschnittlich 5,7 Flexeme pro Lexem existieren. Durch Frequenzuntersuchungen versuchte man nun zu einem hieb- und stichfesten Grundwortschatz zu kommen, der für alle Sprachsituationen gilt. Dieses Unterfangen haben die Linguisten inzwischen wieder aufgegeben. Zu sehr hängt es von der Art des zugrunde gelegten Textes ab, welche Wörter die häufigsten sind. Faßt man die normalsten Alltagssituationen zusammen: Wohnen, Essen, Verkehr und Gesundheit, kommt man in allen Sprachen auf die Zahl 2.000. Dies stellt den Minimalwortschatz dar.

Laut [Zim90] verfügt ein „einigermaßen“ sprachgewandter Deutscher über einen passiven Wortschatz von einigen hunderttausend Wörtern plus dem jeweiligen Fachvokabular seines Berufes. Als Faustregel gilt: Wer eine Fremdsprache lernt, muß mindestens 8.000 Wörter verstehen und 2.000 selbst zu gebrauchen wissen, um sich im Alltag einigermaßen durchschlagen zu können. Umfangreichste Schriftsteller-Wortschätze übersteigen selten 20.000 Wörter. Der aktive Wortschatz eines Deutschen (= Zahl der Wörter, die ihm über Zunge und in die „Feder“ gehen) liegt zwischen 2.000 und 20.000. Der eines durchschnittlich Beredten vermutlich in der Mitte bei 10.000.

Bloße drei Wörter (*der*, *die* und *und*) machen 10% jedes normalen Textes aus, dies gilt auch für die englische Sprache und die Wörter *the*, *a* und *and*. Weitere 66 Wörter decken die Hälfte jede normalen Textes ab. Diese sogenannten *Funktionswörter*: Pronomen, Artikel, Konjunktionen, Präpositionen, Hilfsverben, Interjektionen und Pronominalverbien sind Wörter, die selbst keine oder unscheinbare lexikalische Bedeutung haben, aber klarstellen, in welchen Beziehungen die „Inhaltswörter“ eines Satzes zueinander stehen.

Ungefähr rund 4.000 Wörter machen 75-80% eines normalen unspeziellen, also „leichten“ Textes aus. Um 95% eines fachlich nur schwach spezifizierten, also „normalen“ Textes erschließen zu können, sind jedoch schon 50.000 Wörter nötig. Die letzten 2% eines Textes sind sogar solch unvorhersehbare Wörter, daß man sämtliche Fachwörterbücher zusammennehmen müßte, um den größten Teil von ihnen aufzuführen. Daher ist es nahezu unmöglich, das notwendige Sprachmodell und den Wortschatz eines sprachverarbeitenden

¹ Kleinste bedeutungstragende Einheit eines Wortes.

² Grundform eines Wortes, wie es in konventionellen Wörterbüchern verzeichnet steht.

³ Beugungen.

Systems von vornherein zur Verfügung zu stellen. Vielmehr ist es sinnvoller die erforderliche Sprache eines solchen Systems im Laufe der Anwendung aufzubauen, um so besser auf die jeweilige Aufgabenstellung eingehen zu können.

4.2 Adaptiver Erwerb von Sprache

Die Architektur eines Netzes beeinflusst in hohem Maße seine Leistungsfähigkeit und sein Leistungsvolumen. Aus diesem Grund ist es von großer Bedeutung, eine der Problemstellung angemessene Architektur auszuwählen. Die von Gorin in [Gor90, Gor91A] vorgeschlagene Architektur ist eine akzeptable Alternative, um die Anforderungen zu erfüllen, die die hier gegebene Aufgabenstellung an ein textverarbeitendes System stellt.

Bei Problemstellungen, die sich mit der Verarbeitung natürlicher Sprache befassen, stützt sich die traditionelle Vorgehensweise auf ein vordefiniertes Vokabular und bestimmte vorgegebene Syntaxregeln. Im Gegensatz zu diesem Ansatz beschreibt Gorin ein System, das sich sein Sprachmodell für eine bestimmte Aufgabe aus Information der semantischen Ebene automatisch selbst erwirbt. Die Idee des Systems ist, gesprochene bzw. geschriebene Eingabe in eine Maschinenaktion umzusetzen. Ein neuartiger adaptiver Trainingsalgorithmus schätzt und bestimmt die Gewichte der Verbindungen. Dieser Trainingsalgorithmus vereinigt die Vorteile eines sehr schnellen und, in bezug auf die Reihenfolge der Eingabe, invarianten Lernens, das nur einen einzigen Durchlauf durch das ganze Netzwerk benötigt.

Das ursprüngliche Modell in [Gor90] wird durch ein Einschichten-Netzwerk realisiert. Die Elemente der Eingabeschicht des Netzwerkes repräsentieren das Vokabular des Systems. Für jedes neue Wort, das vom System in seiner geschriebenen bzw. gesprochenen Eingabe angetroffen wird, wird ein neuer Knoten erzeugt und so das Sprachmodell adaptiv erweitert. Dieses Netzwerk geht nicht auf die Reihenfolge der Wörter innerhalb eines Textes ein. Gorin bezeichnet es auch als *bag of words*-System. Die Eingabe des Netzwerkes ist somit unbeschränkter Text aus einem Vokabular von m Wörtern. Ausgabe ist eine von l semantischen Aktionen. Jeder Eingabeknoten stellt einen Worterkenner dar, wobei der n -te Knoten den Eingabesatz s durchläuft und eine Ausgabe O_n erzeugt. Die Ausgabe schwankt zwischen 0 und 1, je nachdem ob sich das n -te Wort des Vokabulars im Satz s befindet. Der Worterkennungsalgorithmus kann dabei willkürlich gewählt werden. Eine Möglichkeit ist ein Erkenner für perfekte Übereinstimmung. Dabei wird ein Ergebnis von 1 erzeugt, falls das betreffende Wort im Satz s angetroffen wird, anderenfalls 0. Bild 4.1 stellt das beschriebene Netzwerk graphisch dar.

Das Gewicht der Verbindung zwischen dem n -ten Wort des Vokabulars und der k -ten semantischen Aktion wird als Verbindungsgewicht w_{nk} bezeichnet. Die Aktivierung jeder semantischen Aktion wird dann durch die Standardformel

$$a_k = \arctan \left[\sum_{n=1}^m O_n w_{nk} \right]$$

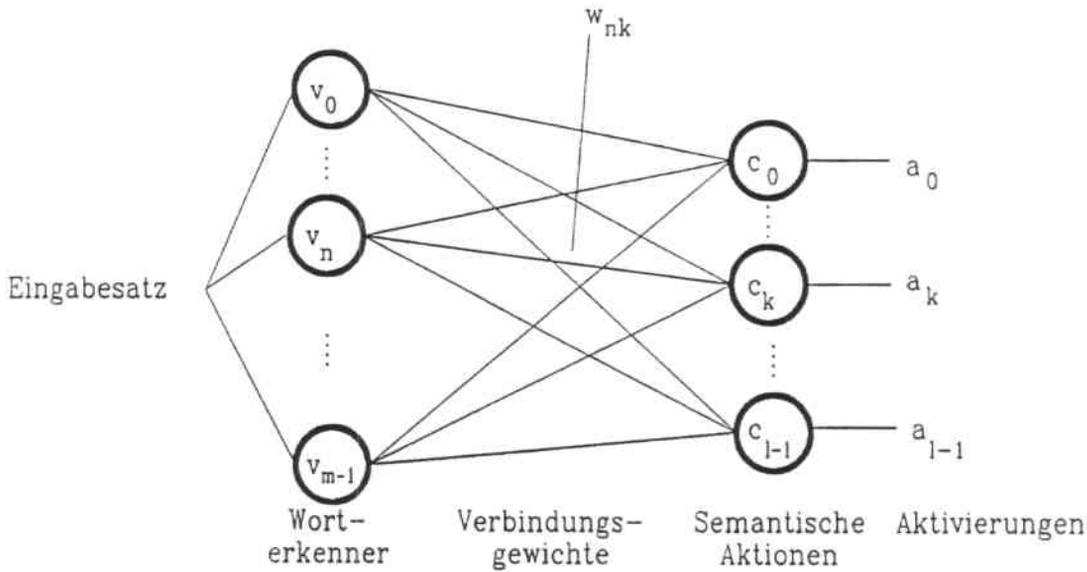


Abbildung 4.1: Abbildung eines Satzes auf seine Bedeutung

berechnet. Der Eingabesatz wird so verstanden als bedeute er die semantische Aktion c_k , deren Aktivierung a_k maximal ist. Dabei wird eine neuartige Berechnung der Verbindungsgewichte benutzt wie sie von Gorin und Levinson vorgeschlagen wurde:

$$w_{nk} = I(c_k, v_n) = \log \frac{P(c_k | v_n)}{P(c_k)},$$

wobei $I(c_k, v_n)$ die wechselseitige Information zwischen dem einzelnen Wort v_n und der semantischen Aktion c_k darstellt. Wechselseitige Information ist im Sinne von Wahrscheinlichkeiten definiert, die nacheinander durch Zähler und Glättungsparameter geschätzt werden. Dadurch ist der Trainingsalgorithmus sequentiell. Gorin und Levinson haben gezeigt, daß diese informationstheoretische Anpassung die Summe der Fehlerquadrate garantiert abnehmen läßt. Der neuartige Trainingsalgorithmus weist, verglichen mit Gradientenabstieg, noch einige andere Vorteile auf:

1. Das Netzwerk lernt in einem einzigen Durchlauf und kann deshalb schnell an verschiedene Aufgabenstellungen angepaßt werden.
2. Es wird keine explizite Auswahl der Schrittgröße des Lernens benötigt.
3. Das Training ist reihenfolgeunabhängig.
4. Es findet eine garantierte Konvergenz zu einer einzigen globalen Parametermenge statt, ein Steckenbleiben in lokalen Minima kann nicht erfolgen.

In herkömmlichen Systemen handelt es sich zumeist um Systeme, die sequentiell und überwacht lernen (siehe auch Abschnitt 3.5). Bei jedem Schritt wird dem System ein Muster und ein Etikett für die richtige Klasse dieses Musters vorgegeben. Dabei wird vom klassifizierenden Teil des Systems ein Fehlersignal erzeugt, das die Abweichung des Eingabemusters vom idealen Muster angibt. Dieses Fehlersignal, zusammen mit der richtigen Klasse des Eingabemusters wird benutzt, um die Parameter des Systems zu verändern. Diese Vorgehensweise ist in Bild 4.2 illustriert.

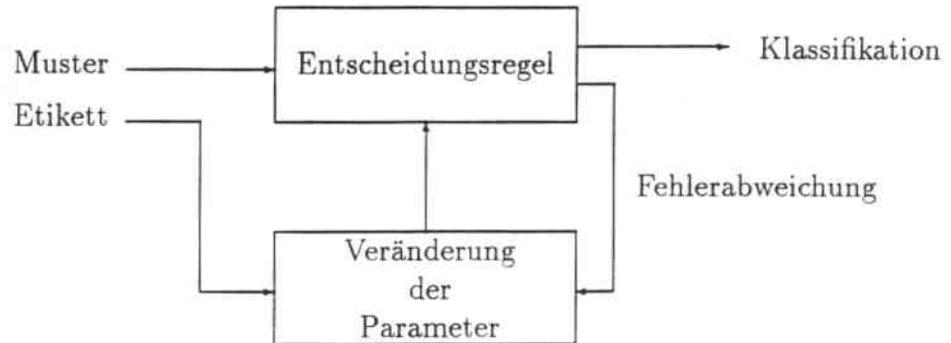


Abbildung 4.2: Traditionelles überwachtes Lernen

Bei der in [Gor90, Gor91A, Mil91] vorgeschlagenen Vorgehensweise wird das Fehlersignal, das die rückkoppelnde Lernschleife kontrolliert, auf semantischer Ebene erzeugt. Nachdem das System als Antwort auf ein Eingabemuster eine Aktion ausgeführt hat, agiert diese Aktion mit dem Benutzer des Systems, der Umwelt oder möglicherweise anderen Systemen. Dadurch wird ein Fehlersignal erzeugt, das die Korrektheit der Antwort anzeigen soll. Der Ablauf dieser Art des Lernens wird in Bild 4.3 veranschaulicht.

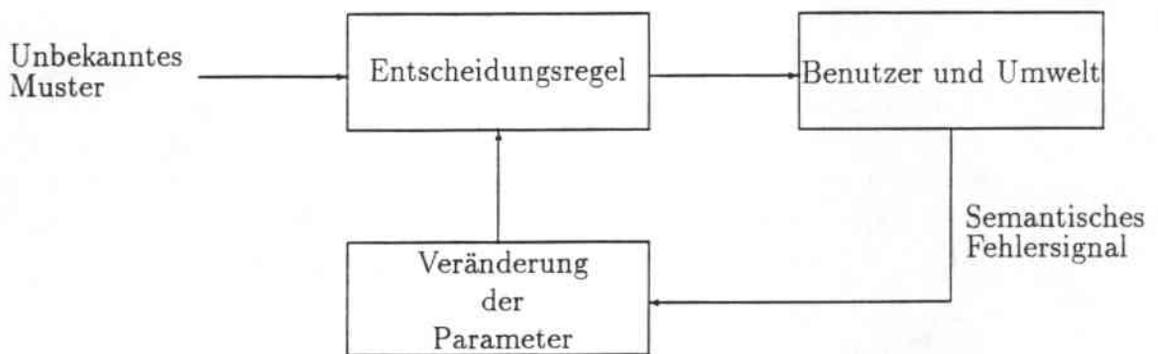


Abbildung 4.3: Rückkopplung der Fehlermeldung auf semantischer Ebene

Die zwei Hauptunterschiede zwischen dem traditionellen und diesem Ansatz sind also folgende:

1. Das Fehlersignal wird auf einer anderen höheren Ebene gegeben. Dabei wird semantische Richtigkeit beurteilt.

- Die richtige Klassifizierung des Eingabemusters (hier: einer Email-Nachricht) muß durch die Interaktion mit der Umwelt abgeleitet werden, während sie in einem überwachten System schon von vornherein zur Verfügung gestellt wird.

Im Einschichten-Modell wird die Reihenfolge der Wörter eines Eingabetextes in das System nicht berücksichtigt. Das Modell ist jedoch auf ein Mehrschichten-Modell erweiterbar. Es wird eine Zwischenschicht eingeführt, die *Phrasen*, in diesem Fall [Gor90] Wortpaaren, entspricht. Die Knoten dieser Zwischenschicht erzeugen, wie schon die Eingabeknoten, eine Ausgabe von 0 oder 1. Diese Ausgabe ist abhängig davon, ob das repräsentierte Wortpaar $v_n v_m$ im Eingabetext erkannt wird. Die Verbindungen der ersten Schicht zur Ausgabeschicht bleiben bestehen. Die Verbindungsgewichte der Zwischenschicht werden mit w_{nmk} bezeichnet und als die wechselseitige Information der „Phrase“ $v_n v_m$ mit den semantischen Aktionen definiert:

$$\begin{aligned} w_{nmk} &= I(c_k, v_n v_m) - I(c_k, v_n) - I(c_k, v_m) \\ &= \log \frac{P(c_k | v_n v_m)}{P(c_k)} - \log \frac{P(c_k | v_n)}{P(c_k)} - \log \frac{P(c_k | v_m)}{P(c_k)}. \end{aligned}$$

Dadurch wird die Leistung des Systems bei Aufgaben, bei denen die Reihenfolge der Wörter eines Textes eine Rolle spielt, verbessert. Es kann jedoch gezeigt werden, daß das Mehrschichten-Netzwerk bei sehr einfachen Aufgaben, ohne Relevanz der Wortfolge, zum Einschichten-Modell konvergiert. Eine Darstellung eines solchen Mehrschichten-Netzwerkes findet sich in Kapitel 5, das das hier implementierte ACEPS vorstellt.

Die neuartigen technischen Ideen, die innerhalb des Systems verwirklicht wurden und die Brauchbarkeit und Leistung des Systems an sich wurden anhand drei verschiedener Aufgabenstellungen bewertet (siehe [Gor90]). Alle Aufgabenstellungen wurden jedoch nur in einem begrenzten Umfang getestet. Bei keinem der durchgeführten Experimente enthielt das Vokabular, das sich das System im Laufe der Aufgabe erwarb, nach Beendigung mehr als 1.600 Worte. Die gesprochene bzw. geschriebene Eingabe wurde in Maschinenaktionen umgewandelt und dabei in zwei, drei und bei einem Experiment sogar in 15 verschiedene Kategorien klassifiziert. Dabei erreichte das Netz Genauigkeitsraten von 66% bis zu 100%, abhängig von der Länge der Eingabe, dem Aufbau des Netzes und der Menge der Trainingsdaten. Die Genauigkeiten, mit der Trainingsdaten richtig klassifiziert wurden, lag zwischen 92% und 100%. Vor allem der Aufbau des Netzwerkes war von Bedeutung: Ein Netz mit mehreren Schichten erzielte bessere Ergebnisse als ein Netz mit nur einer einzelnen Schicht.

Auf diesem Modell basiert die Entwicklung des ACEPS. Durch den Einsatz des ACEPS für Nachrichten, die durch Elektronische Post befördert werden, handelt es sich um eine Aufgabenstellung, bei der die verwendete Sprache nur schwer eingeschränkt werden kann. Der Inhalt der zu klassifizierenden Texte und damit die verwendete Sprache ist nicht vorhersehbar. Deshalb verfügt das im Rahmen dieser Arbeit implementierte ACEPS über kein vordefiniertes Vokabular und keine vorgegebenen Grammatikregeln, sondern baut sich selbst adaptiv ein immer größer werdendes Vokabular auf. Ebenso wird das Fehlersignal, das bei der Klassifikation eines Eingabemusters entsteht, auf semantischer Ebene erzeugt.

4.3 Morphologie

Wie bereits in Abschnitt 4.2 beschrieben, wird in einem Netzwerk, das sich adaptiv seinen Sprachumfang erwirbt, die grammatikalische Struktur der Sprache nicht mit berücksichtigt. Der Inhalt einer Email wird lediglich als *bag of words* betrachtet, wobei jedes Element in der Eingabeschicht des Netzwerkes ein solches Wort repräsentiert. Da sich der Wortschatz eines Menschen bei natürlich sprachlicher, freier Kommunikation nur schwer beschränken läßt, wurde auch in dieser Arbeit keine Einschränkung des zulässigen Vokabulars innerhalb der zu verarbeitenden Email vorgenommen. Dennoch ist es möglich, die Anzahl der Eingabeelemente des Netzwerkes zu reduzieren. Betrachtet man nur die semantische Grundbedeutung eines Wortes, so ist die genaue grammatikalische Form von sekundärem Interesse. Statt z.B. fünf Eingabeelementen für die fünf Wörter *learn*, *learned*, *learning*, *learns* und *learner* gibt es dann nur noch ein einziges Element, das die gemeinsame Grundform der Wörter, nämlich *learn* darstellt. Morphologie, die Lehre von der Bildung von Wortstämmen und ihrer Beugung, ermöglicht die Bildung dieser kleinsten bedeutungshaltigen Einheiten einer Sprache, der Morpheme. Der Wortschatz des ACEPS verkleinert sich bei ausschließlicher Berücksichtigung von Morphemen in den durchgeführten Simulationen um 10-11%. Eine genaue Beschreibung dieser Experimente findet sich in Kapitel 6.

Bei dem innerhalb dieser Arbeit implementierten ACEPS wurde das von Hausser [Hau92] entwickelte LA-MORPH System zur Reduzierung aller Wörter innerhalb der Emails auf ihre Morpheme verwendet. Die Theorie und Methodologie dieses Systems ergaben sich beim Aufbau eines umfassenden englischen Lexikons zum Parsen großer Datenbanken. Angesichts dieser Anwendung war es sinnvoll, die morphologische Analyse der englischen Sprache auf orthographische anstatt der üblichen phonetischen Repräsentation zu stützen. Trotzdem ist LA-MORPH auch ebenso auf gesprochene Sprache anwendbar. Außerdem ist der Einsatz nicht auf die englische Sprache beschränkt, sondern wurde bisher ebenfalls erfolgreich auf Latein, Deutsch, Französisch, Spanisch und Japanisch angewandt.

Morphologie mit Hilfe von Rechnern analysiert Wortformen zum Zwecke der Verarbeitung natürlicher Sprache. LA-MORPH stellt einen neuen Ansatz in der morphologischen Analyse dar. Es basiert auf dem Algorithmus einer links-assoziativen Grammatik, der bis zu diesem Zeitpunkt nur zu syntaktisch-semantischer Analyse natürlicher und formaler Sprachen angewendet wurde. Abgesehen von diesem links-assoziativen Algorithmus unterscheidet sich LA-MORPH von anderen Worterkennungssystemen auch dadurch, daß die Generation von Allomorphen⁴ von der Verkettung der Morpheme getrennt ist. Die Allomorphe werden automatisch während des Ladens des Lexikons erzeugt, die Konkatenation von Morphemen findet während der Laufzeit statt. Dieser Entwurf führt zu optimierter Platzeffizienz bei der Speicherung von Wörtern und einem zeiteffizienten Nachschlagen von Wortformen. Das Hinzufügen neuer Wörter in das Lexikon wird erleichtert, ebenso das Schreiben von Grammatiken für die morphologische Analyse.

⁴Allomorphe sind in gewissem Sinn „erweiterte“ Formen von Wörtern des englischen Wortschatzes. Dabei werden bestimmte Regeln angewandt, um die kleinste Grundform und eventuell existierende unregelmäßige Formen zu erhalten. Beim Substantiv *speaker* z.B. werden die Allomorphe *speak* und *spoke* erzeugt.

Das LA-MORPH System gliedert sich in zwei Teile: das Lexikon und das sogenannte Allomorph-Lexikon. Das Lexikon enthält 8.000 Einträge. Jeder Eintrag besteht aus der Grund- oder Stammform eines Wortes und seiner Kategorisierung. Unter Kategorisierung versteht man die Wortart (Substantiv, Verb,...) und die Klasse innerhalb dieser Wortart. So werden englische Substantive in verschiedene Plural-Klassen eingeordnet. Die Mehrzahl eines Hauptwortes kann durch einfaches Anfügen eines *-s* gebildet werden (*house* → *house-s*) oder kann unregelmäßig erfolgen (*woman* → *women*). Englische Verben wiederum können nach verschiedenen Formen der Vergangenheitsbildung unterschieden werden. Regelmäßige Verben erhalten ein *-d* als Endung (*receive* → *receive-d*), andere verfügen über unregelmäßige Formen (*take* → *took*). Das Allomorph-Lexikon enthält „erweiterte“ Formen der im normalen Lexikon abgespeicherten Grundformen. Zur Grundform *sleep* enthält das Allomorph-Lexikon die Grundform *sleep* und die Vergangenheitsform *slept*, zum Eintrag *speaker* findet man die Formen *speak* und *spoke*. Damit wird das im Arbeitsspeicher befindliche Lexikon um ca. 20% vergrößert.

Die Vorgehensweise des Systems bei der Analyse eines Wortes ist dabei folgende: Das Wort *speaks* wird dem System übergeben. Im Allomorph-Lexikon wird die kleinste Grundform, das Morphem, von *speaks* gesucht. Dann wird mit Hilfe des Eintrags im normalen Lexikon und den möglichen Endungen der dem Wort zugehörigen Klasse versucht, die größtmögliche Form zu bilden, die dem gesuchten Wort entspricht. LA-MORPH liefert selbstverständlicherweise nicht nur das Morphem eines eingegebenen Wortes, sondern analysiert auch seine exakte grammatikalische Form. Ausgabe ist eine Liste aller möglichen grammatikalischen Formen sowie des Morphems, aus dem diese Form entstanden ist. Im oben erwähnten Beispiel ist dies eindeutig, es handelt sich bei *speaks* um die 3. Person Singular der Grundform *speak*. Das Wort *speak* selbst jedoch kann sowohl den Infinitiv, die 1. bzw. 2. Person Singular als auch die 1., 2. oder 3. Person Plural von *speak* darstellen.

Im bisherigen Stadium des ACEPS wurde diese Analyse noch nicht weiter berücksichtigt. Für weitere Anwendungen ist eine sinnvolle Verarbeitung und Mitberücksichtigung der grammatikalischen Form eines Eingabewortes jedoch durchaus sinnvoll.

4.4 Glättungstechniken

Eines der Hauptprobleme stochastischer Sprachmodelle, zu denen sich auch das Sprachmodell des ACEPS zählen läßt, ist die Größe des Trainingskorpus. Um eine Maximum-Likelihood Schätzung [Mül91] vornehmen zu können, reicht die Anzahl der Trainingseinheiten für das jeweilige Modell beinahe niemals aus. Eine Lösung des Problems ist der Versuch, die Parameter des Modells zu glätten. Dies kann durch *Cooccurrence Smoothing* erreicht werden. *Cooccurrence Smoothing* wurde durch Sugawara [Sug85, Sug86] eingeführt.

In [Ess92, Ney91] wird *Cooccurrence Smoothing* zur Spracherkennung verwendet. Bei der dort vorgestellten *Smoothing* Methode wird eine Schätzung der Konfusionswahrscheinlichkeiten von Wortpaaren vorgenommen. Die Wortpaare bestehen aus den Worten *w* und *w'*, die sich „ähnlich“ verhalten. Ähnlichkeit wird dabei als die Wahrscheinlichkeit definiert, mit der das Wort *w* im selben Kontext angetroffen werden wird, in dem *w'*

beobachtet wurde. Diese Konfusionswahrscheinlichkeiten führen zu einer Konfusionsmatrix, die dazu benutzt wird, die bedingten Wahrscheinlichkeiten der einzelnen Wörter zu glätten. Wird Spracherkennung mit sehr großem Wortschatz betrieben, ist es sehr schwierig, eine ähnlich gute Performanz zu erreichen, wie man sie in Systemen mit beschränktem Vokabular vorfindet. Das liegt zum größten Teil daran, daß die Anzahl der Trainingsdaten im allgemeinen nur sehr klein ist, verglichen mit der Anzahl der freien Modellparameter. Viele Merkmale, die in späteren Daten vorkommen, können durch die nur geringe Anzahl von Trainingsdaten gar nicht erfaßt werden. Folglich kann auch das Netz nicht dahingehend trainiert werden, sie zu erlernen und später zu erkennen. Durch eine Maximum-Likelihood Schätzung würden solche nicht im Training angetroffenen Beobachtungen eine Wahrscheinlichkeit von Null erhalten. Ein Prinzip der Spracherkennung ist aber, daß keine Wortsequenz von der Erkennung ausgeschlossen sein soll, deshalb müssen Wahrscheinlichkeiten von Null vermieden werden. Durch eine Glättung der Sprachparameter kann eben dieses Ziel erreicht werden. Mit Hilfe von Cooccurrence Smoothing gelang es, die Perplexität eines 1.000.000 Wörter großen Textkorpus in englischer Sprache um 14,4% zu verbessern.

Cooccurrence Smoothing wird auch bei sprecherunabhängiger Phonemerkennung [Lee88A] eingesetzt. Dort ergibt sich ebenfalls das Problem einer begrenzten Menge von Trainingsdaten. Cooccurrence Smoothing bestimmt die Ähnlichkeit zwischen allen Paaren von Codewörtern aller Phoneme und glättet die individuellen Verteilungen dementsprechend. Dabei wurden gute Ergebnisse schon mit 16 gesprochenen Trainingssätzen von nur zwei Sprechern erzielt, während vorher 2.830 Sätze von 357 Sprechern notwendig gewesen waren.

4.5 Backpropagation

Leistung und Repräsentationsmöglichkeiten neuronaler Netzwerke hängen vor allem von der Anzahl der Schichten, die „lernen“, ab. Ein einlagiges rückkopplungsfreies Netzwerk besteht nur aus Eingabe- und Ausgabelementen und enthält keinerlei verdeckte Elemente. Die Menge an der Eingabeschicht ankommender Eingabemuster wird direkt auf eine Menge von Ausgabemustern abgebildet. Ähnliche Eingabemuster erzeugen dabei ähnliche Ausgabemuster. Dieses Verhalten führt zu einer wichtigen Eigenschaft einlagiger rückkopplungsfreier Netze: Sie sind in der Lage, vernünftige Generalisierungen auszuführen und können damit auch auf unbekannte, nicht gelernte Eingaben eine „richtige“ und angemessene Ausgabe liefern. Für eine ganze Reihe von Problemen ist es jedoch nicht ausreichend, ähnliche Eingabevektoren auf ähnliche Ausgabevektoren abzubilden. Es handelt sich dabei um die Klasse nicht linear trennbarer Probleme, die nicht durch einlagige rückkopplungsfreie Netzwerke lösbar sind.

Ein immer wieder zitiertes klassisches Beispiel ist das Exklusive-Oder (XOR) Problem. Hier sollen gerade die sich am wenigsten ähnlichen Eingabemuster 0/0 und 1/1, siehe auch Tabelle 4.1, eine identische Ausgabe, nämlich 0, liefern. Wie viele andere Probleme dieser Art kann diese Aufgabenstellung nicht durch ein Netzwerk ausgeführt werden, das keine interne Repräsentation der Eingaben ermöglicht. Verdeckte, lernfähige Schichten jedoch sind zu solchen internen Repräsentationen fähig. Repräsentations- und Pro-

Eingabemuster	Ausgabemuster
0 0	0
0 1	1
1 0	1
1 1	0

Tabelle 4.1: XOR-Problem

blemlösungsfähigkeit eines Mehrschichten-Netzes hängen immer von der Zahl der Schichten und der Anzahl der Elemente pro Schicht ab. So kann das XOR-Problem z.B. durch Hinzufügen nur eines verdeckten Elements in ein Netzwerk mit zwei Eingängen, die mit einem Ausgang verbunden sind, gelöst werden.

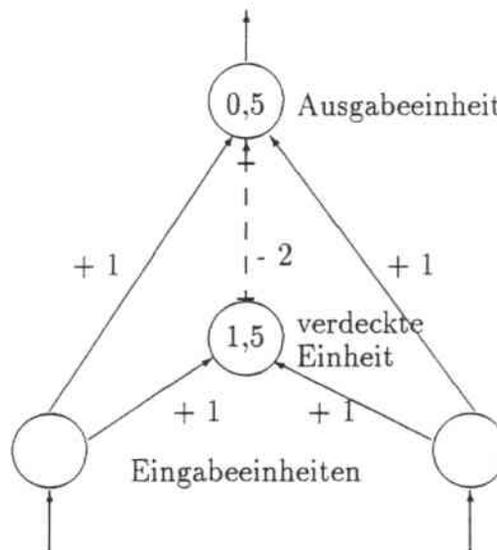


Abbildung 4.4: Netzwerk zur Repräsentation des XOR-Problems

Bild 4.4 stellt ein Netzwerk zur Lösung des XOR-Problems dar. Die Zahlenangaben an den Pfeilen repräsentieren die Gewichte der Verbindungen zu den Elementen. Die Zahlen innerhalb der Kreise geben die Schwellwerte der einzelnen Elemente an. Der Schwellwert $+1,5$ des verdeckten Elements sorgt dafür, daß dieses Element nur dann aktiv ist, wenn auch beide Eingabeelemente aktiv sind. Durch das hohe negative Gewicht des verdeckten Elements zum Ausgabeelement ist dieses nur dann aktiv, wenn auch beide Eingabeelemente in diesem Zustand sind. Aus der Sicht des Ausgabeelements betrachtet, kann das verdeckte Element somit als dritte Eingabe betrachtet werden.

Angesichts solcher Netzwerke wird die Stärke verdeckter Elemente und interner Repräsentationen deutlich. Ein Problem bleibt dabei immer noch die Anpassung der Gewichte in den verborgenen Schichten. Während für Netzwerke ohne verdeckte Schichten Lernregeln existieren, gab es lange Zeit kein entsprechendes Gegenstück für Netzwerke mit verdeckten Schichten. Der Backpropagation-Algorithmus stellt mit der *generalisierten Delta-*

Regel einen Adaptionsmechanismus für beliebige, rückkopplungsfreie Netzwerkstrukturen zur Verfügung. Er wurde Mitte der 80er Jahre von verschiedenen Arbeitsgruppen unabhängig entwickelt. Allerdings wurde er erst durch eine Veröffentlichung von D. E. Rumelhart, G. E. Hinton und R. J. Williams 1986 [Rum88] bekannt und beeinflusste die weitere Entwicklung neuronaler Netzwerke entscheidend. Die generalisierte Delta-Regel erlaubt die Berechnung eines Fehlersignals auch für die Elemente verdeckter Schichten eines Netzwerkes.

In einem rückkopplungsfreien Backpropagation-Netzwerk verkörpern die Eingabelemente die unterste Schicht, die Ausgabeelemente die oberste Schicht. Dazwischen kann eine beliebige Anzahl verdeckter Schichten liegen. Die Verbindungen zwischen den einzelnen Elementen eines Netzes laufen immer in Richtung von Eingabe zu Ausgabe. Dabei ist es nicht zwingend, daß nur direkt benachbarte Schichten Verbindungen haben, eine Verbindung kann auch beliebig viele Zwischenschichten überspringen. Ein mehrschichtiges Netzwerk mit linearer Aktivierungsfunktion kann auf ein einschichtiges Netzwerk reduziert werden. Dazu werden alle Schichten zu einer Schicht mit geeigneter Gewichtsmatrix zusammengefaßt. Aus diesem Grund wird beim Backpropagation-Algorithmus eine *semilineare* Aktivierungsfunktion verwendet. Eine semilineare Funktion ist eine nicht lineare Funktion, in der die Ausgabe eines Elements eine streng monoton steigende und differenzierbare Funktion der gesamten Netzausgabe darstellt. Die Ausgabe o_j eines Elements j des Netzes berechnet sich nach folgender Formel:

$$o_j = a_j = F_j\left(\sum_i w_{ij} \cdot o_i\right),$$

wobei F_j die semilineare Aktivierungsfunktion von j repräsentiert. Die generalisierte Delta-Regel ist dann anwendbar, wenn das Netzwerk nur aus Elementen besteht, die semilineare Aktivierungsfunktionen haben. Prinzipiell kann also jedes Element des Netzes eine eigene Aktivierungsfunktion haben, in der Regel wird jedoch zumindest für alle Elemente einer Schicht eine einheitliche Aktivierungsfunktion F definiert.

Für den Einsatz der generalisierten Delta-Regel spielt auch die Ableitung der Aktivierungsfunktion eine große Rolle. Das setzt voraus, daß eine Aktivierungsfunktion angewendet wird, für die auch eine Ableitung existiert. Am häufigsten wird daher in Backpropagation-Netzwerken die *Sigmoid-Funktion* (oft auch logistische Funktion genannt) verwendet:

$$\begin{aligned} o_j = a_j = F(\text{net}_j) &= \frac{1}{1 + e^{-g \cdot \text{net}_j}} \\ &= \frac{1}{1 + e^{-g \cdot \left(\sum_i w_{ij} \cdot o_i + \Theta_j\right)}}. \end{aligned}$$

Dabei ist Θ_j ein Bias, der einem Schwellwert entspricht und g ein Parameter, der die Steilheit des Kurvenverlaufs bestimmt, und auch als *gain* bezeichnet wird. Dieser Parameter hat häufig den Wert $+1$, was auch in den folgenden Formeln angenommen wird. Die Sigmoid-Funktion bietet den Vorteil einer einfachen ersten Ableitung:

$$\begin{aligned}
 F'(net_j) &= \frac{\partial F(net_j)}{net_j} \\
 &= F(net_j) \cdot (1 - F(net_j)) \\
 &= \frac{1}{1 + e^{-\left(\sum_i w_{ij} \cdot o_i + \Theta_j\right)}} \cdot \left(1 - \frac{1}{1 + e^{-\left(\sum_i w_{ij} \cdot o_i + \Theta_j\right)}}\right).
 \end{aligned}$$

Auch die Sinusfunktion und der Tangens Hyperbolicus sind häufig eingesetzte Aktivierungsfunktionen in Backpropagation-Netzwerken.

Die Lernregel eines Backpropagation-Netzwerkes geht nun folgendermaßen vor: Das Netz erhält aus eine Menge vorklassifizierter Ein-/Ausgabepaare einen Eingabevektor und produziert mit Hilfe dieses Eingabevektors einen Ausgabevektor. In der Lernphase des Netzes werden die Aktivitäten der Ausgabeelemente mit den gewünschten Ausgaben verglichen und bei Abweichungen werden die Gewichte des Netzes angepaßt. Sind keine Differenzen vorhanden, findet auch kein Lernen statt. Auf diese Weise wird der Fehler der Ausgabe-schicht über alle Schichten bis zur Eingabeschicht zurückpropagiert. Der gesamte Fehler eines Netzwerkes ist der über alle m Musterpaare definierte quadratische Fehler:

$$E = \frac{1}{2} \sum_{k=1}^m \sum_j (z_j^k - o_j^k)^2.$$

Hierbei ist z_j^k die gewünschte Ausgabe des j -ten Elementes des Eingabevektors k , o_j^k ist das j -te Element der durch Verarbeitung des Eingabevektors k entstehenden tatsächlichen Ausgabevektors. Der Fehler eines einzelnen Ein-/Ausgabepaares k ist

$$E_k = \frac{1}{2} \sum_j (z_j^k - o_j^k)^2.$$

E ergibt sich daraus also durch

$$E = \sum_{k=1}^m E_k.$$

Die Delta-Lernregel minimiert diesen Fehler, indem sie die Gewichte immer in Richtung des steilsten Gradienten in Bezug zur Fehleränderung adaptiert. Für alle Elemente des Netzwerkes gilt:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j o_i.$$

dabei gibt η die Lernrate an, mit der das Netz trainiert werden soll und δ_j bezeichnet den Fehler des Elements j .

Die generalisierte Delta-Regel entspricht also im wesentlichen der Standard Delta-Regel. Das Gewicht jeder Verbindung w_{ij} soll einer Summe entsprechend verändert werden, die proportional ist zu dem Produkt des Fehlersignals δ_j , das von dem Element j empfangen wird, und der Ausgabe o_i des Elements i , das seine Aktivierung schickt. Die Bestimmung des Fehlersignals ist ein rekursiver Prozess, der bei den Ausgabeelementen eines Netzwerkes beginnt. Das Fehlersignal δ_j für ein Element der Ausgabeschicht ist folgendes:

$$\delta_j = F'(net_j)(z_j - o_j),$$

wobei z_j der gewünschten Ausgabe des Elements entspricht und o_j die tatsächliche Aktivierung dieses Elements darstellt. $F'(net_j)$ ist dabei die Ableitung der semilinearen Funktion, die als Aktivierungsfunktion verwendet wird. Das Fehlersignal eines verdeckten Elements j wird rekursiv bestimmt, in Form der Fehlersignale aller Elemente zu denen eine Ausgabe des Elements j führt und den Gewichten dieser Verbindungen. Der Fehler δ_j für verdeckte Elemente bestimmt sich folgendermaßen:

$$\delta_j = F'(net_j) \sum_h \delta_h w_{hj}.$$

Dabei wird über alle h summiert, δ_h ist der Fehler der Elemente h , die in der nachfolgenden Schicht liegen.

Die Anwendung der generalisierten Delta-Regel gliedert sich demnach in zwei Phasen: Während der ersten Phase wird die Eingabe eines Trainingsmusters präsentiert und vorwärts durch das Netzwerk verbreitet, um den Ausgabewert o_j jedes Elements zu berechnen. Diese Ausgabe wird dann mit den gewünschten Zielausgaben des Trainingspaares verglichen und ein Fehlersignal δ_j für jedes Element j berechnet. Die zweite Phase umfaßt einen weiteren Durchgang — in umgekehrter Richtung zum Informationsfluß — durch das Netzwerk, währenddessen eine rekursive Propagierung der Fehler δ_j stattfindet. Sie werden zur Berechnung der Fehler vorhergehender Schichten verwendet und die notwendigen Anpassungen der Gewichte werden vorgenommen. Dieser Vorgang wird solange durchgeführt bis die Eingabeschicht erreicht ist und die Fehler aller Elemente berechnet sowie ihre Gewichte verändert sind.

Der gerade beschriebene Lernvorgang soll verständlicherweise nicht zu viel Zeit in Anspruch nehmen. Deshalb wählt man in der Regel eine möglichst große Lernrate, die jedoch nicht zu Oszillation führt. Damit ist ein rasches Lernen gewährleistet. Eine weitere Möglichkeit, den Lernvorgang zu beschleunigen, ohne zu oszillieren, ist die Einführung eines *Momentumfaktors*. Die Berechnung des neuen Gewichts eines Elements berücksichtigt dann zusätzlich die Gewichtsveränderung des vorhergehenden Lernschnittes, d.h. den alten Fehler. Folgende Formel wird hierfür benutzt:

$$\Delta w_{ij}(t) = \eta \delta_j o_i + \mu \Delta w_{ij}(t-1)$$

mit δ_j als Fehler des Elements j zum Zeitpunkt t , o_i als Ausgabe des Elements i zum selben Zeitpunkt t sowie μ als dem Momentumfaktor. Neue Gewichte berechnen sich dann durch:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t).$$

Vorteil dieser Vorgehensweise ist, daß in die aktuellen Gewichtsänderungen auch die Änderungen des vorhergehenden Lernzyklus eingehen. Das Momentum wirkt wie eine Art Tiefpaßfilter. Längerfristig wird damit der richtige Weg gewählt, kurzfristige Schwankungen und „lokale Minima“ werden nicht so stark berücksichtigt.

Das Lernverhalten beim Backpropagation-Algorithmus kann durch verschiedene Faktoren beeinflusst werden. So hängt der Lernerfolg eines Backpropagation-Netzes von der Reihenfolge der dargebotenen Trainingsmuster ab. Die zu lernenden Muster dürfen deshalb nicht in einer festgelegten Reihenfolge, sondern müssen dem Netzwerk zufällig als Eingabe angeboten werden. Damit werden zyklische Veränderungen der Gewichte, die sich nach jedem Lerndurchgang wiederholen, vermieden.

Die Gewichte können zu verschiedenen Zeitpunkten verändert werden:

- sofort nach jedem Lernschritt
(d.h. die Minimierung der Fehlerfunktion erfolgt in Richtung des steilsten Gradienten bezüglich der summierten Fehlerquadrate eines Musters)
- nach einer bestimmter Zahl von Lernschritten
Diese Zahl entspricht meist der Anzahl der zu lernenden Muster. Dabei findet nach jedem Lernzyklus (= Durchlauf aller zu lernenden Muster) eine kumulative Änderung der Gewichte statt, die die Gewichtsänderungen aller Lernschritte für ein bestimmtes Element gespeichert und aufsummiert hat. Die Summe wird dann in einem Schritt zum alten Gewicht addiert.

Letztere Methode hat den Vorteil, daß tatsächlich der Gesamtfehler aller zu lernenden Muster berücksichtigt wird. Außerdem wird er in Richtung des steilsten Gradienten der Gesamtfehlerfunktion minimiert. Des weiteren spielt hier die Reihenfolge der angebotenen Muster keine Rolle mehr, da der Fehler über alle Muster berechnet wird. Die kumulative Methode ist jedoch auch bei nicht sehr großen Trainingsmengen extrem langsam und besonders ungünstig, falls man über eine große Anzahl von Trainingsmustern verfügt. Die Berechnung des Fehlers bei allen Mustern erfordert dann sehr viele Rechenoperationen und die Schnelligkeit der Konvergenz nimmt stark ab.

Auch die an das Netzwerk angelegten Eingabewerte beeinflussen das Lernverhalten des Netzes. Werden binäre Werte 0 oder 1 angelegt, können die Eingänge, an denen Null anliegt, nicht lernen. Grund dafür ist die Änderung des Gewichts w_{ij} , die proportional zu o_j ist. Günstiger ist es also, $\pm \frac{1}{2}$ anzulegen.

Der Lernvorgang eines Backpropagation-Netzwerkes besteht darin, die Fehlerfunktion zu minimieren. Diese Vorgehensweise entspricht einem sogenannten *hill-climbing*-Algorithmus.

Auch dort kann es, bedingt durch lokale Minima, schwierig sein, das globale Minimum zu finden. Der Backpropagation-Algorithmus läuft immer in die Richtung, die aufgrund lokaler Erkenntnis⁵ den steilsten Abstieg bedeutet. Dabei besteht immer die Gefahr, in ein lokales Minimum zu gelangen und dies aufgrund der Fehlerfunktion nicht mehr verlassen zu können. Leider ist es in der Regel nicht möglich, für eine bestimmte Anwendung die Lösbarkeit des Problems und die Konvergenz des Verfahrens vorauszusagen.

Nehmen die Gewichte eines Teils des Netzwerkes sehr große positive oder negative Werte an, kann der Lernvorgang auch sehr behindert werden. Es entstehen hohe positive oder negative Werte für net_j und die Ableitung $F'(net_j)$ der Aktivierungsfunktion ist dann praktisch gleich null. Da die Steigung der Sigmoid-Kurve aber die Gewichtsänderung proportional zur Ableitung der Aktivierungsfunktion ist, kommt das Lernen quasi zum Stillstand. Dieser Entwicklung kann durch eine niedrige Lernrate und einen niedrigen Momentumsfaktor entgegengewirkt werden, was aber eventuell die Lerndauer des Netzes verlängert.

Der Backpropagation-Algorithmus wurde bei den mit dem ACEPS durchgeführten Simulationen ebenfalls verwendet und in die Trainingsprozedur integriert.

⁵Die Ableitung der Fehlerfunktion für gegebene Gewichte ist minimal.

Kapitel 5

Adaptive Connectionist Email Processing System (ACEPS)

Die Motivation zur Schaffung des ACEPS (Adpative Connectionist Email Processing System) wurde bereits in den ersten Kapiteln dieser Arbeit umrissen. Auch die theoretischen Grundlagen der Implementierung eines solchen Systems sind im vorigen Kapitel beschrieben worden. In den folgenden Abschnitten wird nun genauer auf die Realisierung eines textverarbeitenden Systems mittels konnektionistischer Modelle eingegangen, und es findet eine konkrete Beschreibung der implementierten Ideen statt. Dazu werden als erstes die Systemarchitektur und die einzelnen Komponenten des ACEPS vorgestellt. Die Funktionalität dieser Komponenten wird durch die Beschreibung eines Benutzungsablaufs des Systems durch den Menschen erklärt. Schließlich werden die Einsatzmöglichkeiten des ACEPS dargestellt und mögliche Erweiterungen des Systems vorgeschlagen.

5.1 Systemarchitektur

Das ACEPS läßt sich, wie in Bild 5.1 dargestellt, in 7 Komponenten untergliedern. Mittels der Benutzungsschnittstelle tritt ein möglicher Anwender mit dem ACEPS in Verbindung. Durch sie wird der Aufruf der darunterliegenden Komponenten ermöglicht. Die Vorverarbeitung des ACEPS bereitet eine Email so auf, daß die spätere Bearbeitung der Nachricht erleichtert wird. Dies geschieht durch definierte *Shellskripten* [Ker84], die auf C-Programme zurückgreifen. Die vorverarbeitete Eingabe wird entweder durch die Trainings- oder die Testkomponente des ACEPS weiter ausgewertet. Dazu stellt das ACEPS ein Lexikon seines Wortschatzes sowie ein darauf basierendes neuronales Netzwerk zur Verfügung. Beide Komponenten bedienen sich außerdem einer Liste von nicht bedeutungstragenden bzw. nicht deskriptiven oder aussagekräftigen Wörtern, der sogenannten LNW¹ (List of Non-descriptive Words). Sowohl das Lexikon, das Netzwerk, als auch die Liste nicht deskriptiver Wörter werden zwischen den Aufrufen des ACEPS als einfache sequentielle Dateien im Dateisystem des Betriebssystems abgelegt.

¹Dazu gehören Worte wie *a*, *the* oder *and*.

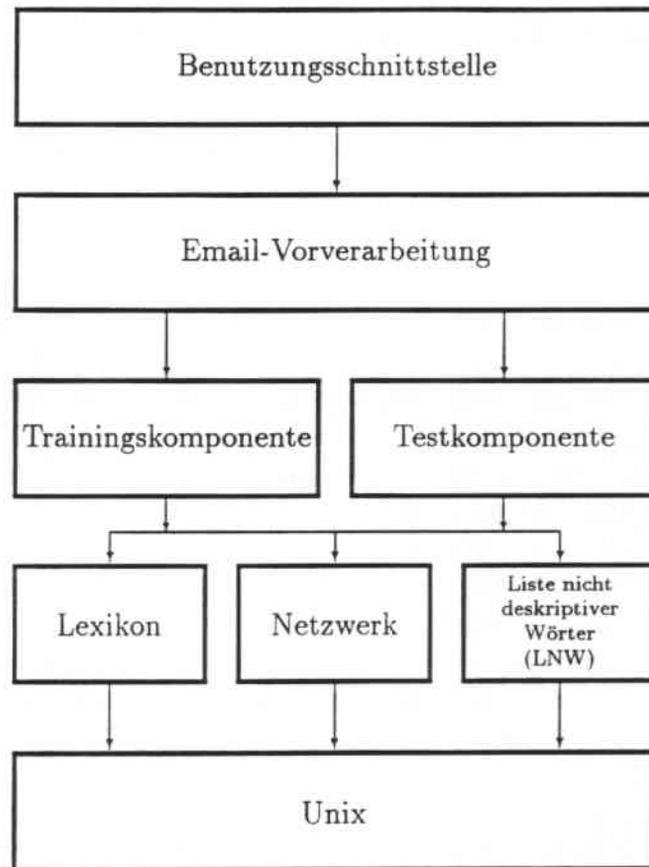


Abbildung 5.1: Systemarchitektur des ACEPS

5.2 Funktionalitätsbeschreibung

Grundlage des ACEPS ist die Idee eines Werkzeuges, das sich mit der Bearbeitung elektronischer Postnachrichten befaßt. Mit Hilfe dieses Email-verarbeitenden Werkzeuges soll die Flut der Nachrichten, die ein Mensch täglich in seinem Briefkasten bzw. seiner *mailbox* vorfindet, gefiltert, vorsortiert und damit einem Benutzer das Lesen seiner Post erleichtert werden.

Die Bearbeitung einer Email durch das ACEPS geht dabei folgendermaßen vor sich: Ein menschlicher Benutzer ruft das ACEPS auf und gibt dabei eine Reihe von Parametern an. Hierbei lassen sich zwei Fälle unterscheiden: Der erste Aufruf des Systems, bei dem die Initialisierung der einzelnen Teilkomponenten des ACEPS automatisch erfolgt, und weitere darauf basierende Aufrufe. Das System erkennt dabei selbständig, ob es sich um die initiale Anwendung handelt. Bei einem Aufruf des ACEPS wird die Identifikation der zu bearbeitenden Email, die Art der Bearbeitung sowie der Bezeichner einer benutzerspezifischen Variante des Systems angegeben. Das ACEPS überprüft, ob ein System dieses Namens bereits existiert. Ist dies nicht der Fall, handelt es sich um den ersten Aufruf und es wird eigens für diesen Benutzer eine Ausprägung des ACEPS geschaffen. Bei seiner erstmaligen Benutzung baut das ACEPS alle notwendigen Strukturen auf. Dazu fordert das System den Benutzer zur Angabe weiterer Parameter auf. So müssen die

Prioritätsstufen, Klassen oder semantischen Aktionen, nach denen das ACEPS seine Eingabe unterscheiden soll, vor Benutzung definiert werden. Damit wird der Rahmen des speziellen benutzerspezifischen Systems bestimmt. Aufgrund dieser Angaben erzeugt das ACEPS intern ein Netzwerk. Die Anzahl der vorgegebenen Klassen gibt dabei die Anzahl der Ausgabeelemente dieses Netzwerkes an. Ebenso findet die Initialisierung eines Lexikons statt, in dem später alle bisher vom ACEPS angetroffenen Wörter abgespeichert werden.

Bei jeder weiteren Benutzung des ACEPS muß lediglich die zu bearbeitende Email sowie die Art der Bearbeitung und der Name des schon vorher erzeugten benutzerspezifischen Systems vom Anwender angegeben werden. Das ACEPS speichert nach jeder Benutzung seinen momentanen Zustand und sein Wissen ab und ist somit bei jedem neuen Aufruf fähig, früher gelernte Tatsachen zu verwerten. Dieses Wissen besteht zum einen aus dem bis zu diesem Zeitpunkt aufgebauten Netzwerk und seinen Gewichten. Zum anderen baut sich das ACEPS im Laufe der Zeit ein Lexikon bereits „gelernter“ Wörter auf. Bei einer Zweitbenutzung werden das in Betriebssystem-Dateien gespeicherte Netzwerk und das Lexikon wieder im ACEPS aufgebaut. Ebenso wird bei der ersten und jeder weiteren Anwendung des Systems die Liste nicht deskriptiver Wörter (LNW) eingelesen. Sie enthält Wörter der englischen Sprache, die in einem hohen Prozentsatz englischer Texte vorkommen. Dabei handelt es sich um Wörter wie *a*, *an*, *the*, *and*, *for*, *with*, die Buchstaben *a* bis *z*, die Zahlen 1 bis 100 und verschiedene Sonderzeichen. Durch die Häufigkeit ihres Auftretens erlauben solche Wörter keinerlei Aussagen über Inhalt oder Themengebiet eines Textes. Die nicht bedeutungstragenden Wörter im Text zu belassen, würde nur bedeuten, daß das Lexikon und Netzwerk des ACEPS unnötig aufgebläht werden, ohne daß sich dadurch die Klassifikation der Texte bzw. die Leistung des Systems verbessert. Die Liste nicht deskriptiver Wörter wurde unabhängig von der Anwendung des ACEPS empirisch ermittelt und wird vom System während seines Ablaufs nicht verändert. Sie kann jedoch mit einfachen Betriebssystem-Kommandos an die Wünsche des Benutzers angepaßt werden. Nachdem nun alle notwendigen Strukturen vorhanden sind, kann die Bearbeitung der Email beginnen.

Zunächst wird eine ankommende Email-Nachricht durch eine Vorverarbeitungskomponente des ACEPS um die zur Verarbeitung unnötige Leitweginformation im Kopf der Nachricht gekürzt. Erhalten bleiben Absender („From“-Zeile) und Gegenstand („subject“-Zeile) der Nachricht, da beide Teile Informationen enthalten können, die für den Inhalt einer Nachricht von Bedeutung sind. Die ursprünglich eingegebene Postnachricht bleibt bei Wegnahme der Leitweginformationen unverändert erhalten. Es wird lediglich eine weitere, veränderte Version dieser Datei geschaffen. Sie trägt den Suffix `.short` und existiert nur temporär während der Verarbeitung durch das ACEPS. Beispiele für eine Email-Nachricht in Originalform sowie nach der Bearbeitung durch die Vorverarbeitungskomponente finden sich in Anhang A.1 und A.2. Abgesehen von dieser Reduzierung durch Wegnahme unnötiger Verwaltungsinformation wird die Nachricht in ihrer Originalform dem ACEPS zur Auswertung übergeben. Das ACEPS kann ankommende Email-Nachrichten auf zwei verschiedene Arten auswerten. Je nach Benutzerwunsch wird die Nachricht als Testmail in der

- Ausführungsphase

oder als Trainingsmail in der

- Lernphase

benutzt. Eine Ausführungs- oder Testphase des Systems ist in Bild 5.2 dargestellt und läuft folgendermaßen ab:

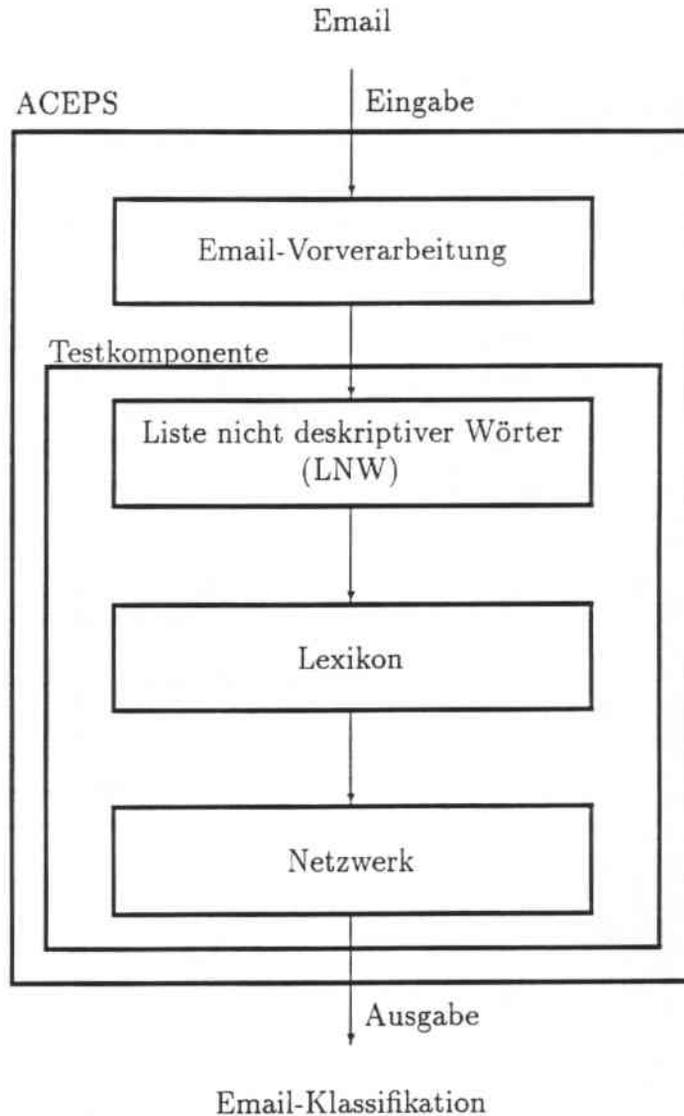


Abbildung 5.2: Funktionaler Ablauf während der Verarbeitung einer Email

Die Testkomponente des ACEPS übernimmt die eingegebene Email und filtert anhand der Liste nicht deskriptiver Wörter (LNW) alle nicht aussagekräftigen Wörter der Nachricht heraus. Anhang A.3 zeigt das Beispiel einer Email nach Herausfiltern der nicht deskriptiven Wörter durch die LNW. Mit Hilfe des Lexikons, in dem sich der gesamte

Wortschatz des ACEPS befindet, werden die entsprechenden Elemente des Netzwerkes aktiviert. Das ACEPS ermittelt anhand seiner Auswertungsstrategie den Aktivierungszustand der Ausgabeelemente und ordnet die Nachricht einer Klasse des vom Benutzer vorgegebenen Rahmens zu.

Um jedoch überhaupt in der Lage zu sein, solche Klassifikationen ausführen zu können, muß das ACEPS sich vorher Wissen angeeignet haben. Dies geschieht in einer Lern- oder Trainingsphase. Bild 5.3 zeigt das Vorgehen während der Bearbeitung einer Trainingsmail.

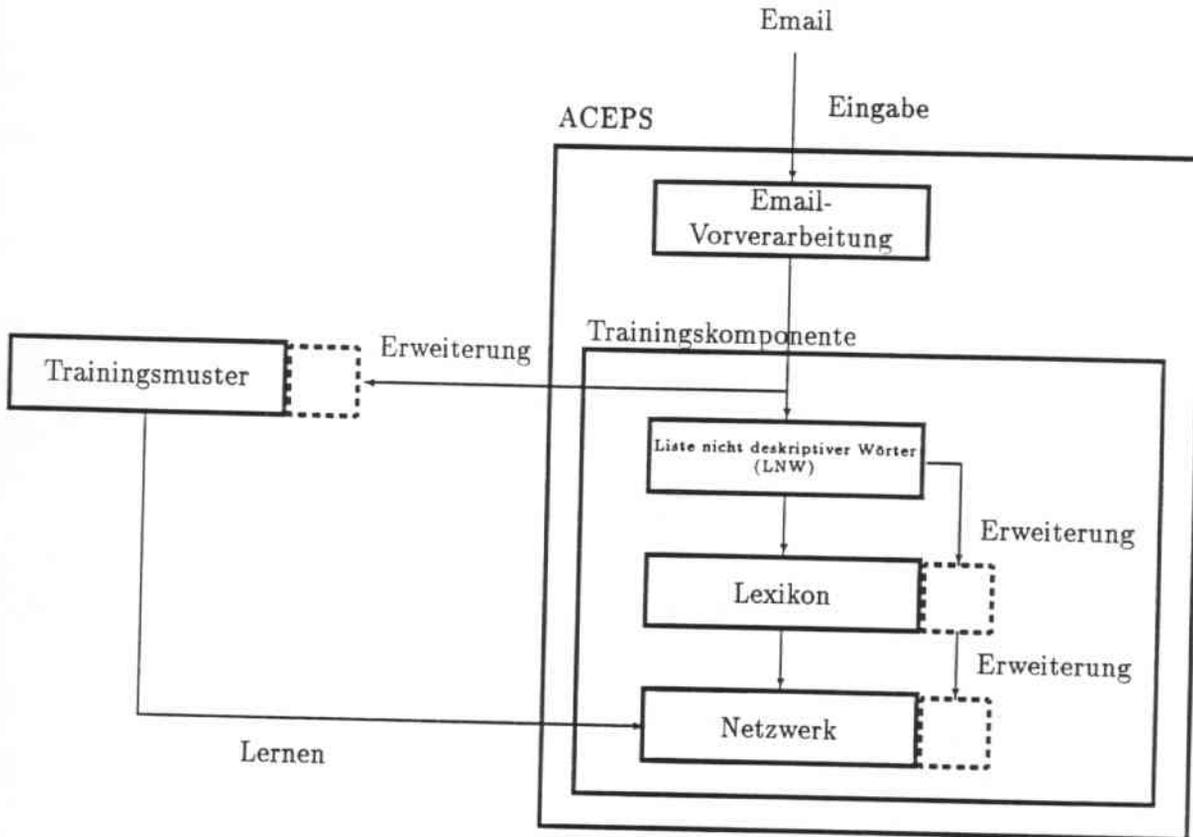


Abbildung 5.3: Funktionaler Ablauf während der Lernphase des ACEPS

Eine Nachricht, die als Trainingsmail verwendet werden soll, muß bereits vorklassifiziert worden sein und ein Etikett oder eine Markierung aufweisen. Nur so kann das neuronale Netzwerk des ACEPS mit Hilfe dieses Ein-/Ausgabepaares neues Wissen erwerben. Diese Art des Lernens wird — wie schon in Abschnitt 3.5 vorgestellt — überwachtes Lernen genannt. Der Lernvorgang läßt sich in zwei Abschnitte unterteilen: Er besteht aus einer Erweiterung der Strukturen des ACEPS mit nachfolgendem Training des Netzes. Als erstes wird die zum Training vorgesehene Email, wie auch „normale“ Testmails, durch die Vorverarbeitungskomponente in geeignete Form gebracht. Ebenso sorgt die Liste nicht deskriptiver Wörter (LNW) für das Herausfiltern nicht relevanter Wörter. Anders als bei einer Klassifikation werden während des Lernens alle dem System unbekannt Wörter, die in der Nachricht vorkommen, in das Lexikon des ACEPS aufgenommen. Gleichzeitig wird das Netzwerk durch Bildung neuer Elemente erweitert. Die Nachricht selbst wird samt Etikett in die Menge der Trainingsmuster eingereiht. Dann kann das

zweite Stadium des Lernens beginnen. Das Netzwerk wird unter Benutzung aller in der Menge der Trainingsmuster vorhandenen Nachrichten trainiert. Durch das Hinzukommen einer neuen Trainingsmail sowie neuer Elemente innerhalb des Netzes und neuer Wörter im Lexikon, müssen die Gewichte neuer Verbindungen bestimmt und die bereits bestehender Verkettungen eventuell neu angepaßt werden. Damit verfügt das System über neues Wissen und ist in der Lage, Emails aufgrund der frisch erworbenen Kenntnisse besser und genauer zu klassifizieren als vorher.

Sind Ausführungs- bzw. Lernphase des ACEPS beendet, werden der aktuelle Zustand von Lexikon und Netzwerk wieder in Betriebssystem-Dateien gespeichert.

5.3 Aufbau und Struktur des Netzwerkes

5.3.1 Entwurf eines Netzwerkes

Grundlagen und Eigenschaften eines neuronalen Netzwerkes wurden bereits in Kapitel 3 erläutert. Sowohl ihr hochgradiger Parallelismus als auch die Ähnlichkeit zur menschlichen Verarbeitungsweise von Information macht die Verwendung neuronaler Netzwerke zur Lösung einer Aufgabe außerordentlich attraktiv.

Damit die endgültige Architektur des Netzes den Anforderungen der Aufgabenstellung entspricht, müssen bei jedem Aufbau eines neuronalen Netzwerkes gewisse Architekturentscheidungen schon vorher getroffen werden. Nach [Her91] ergeben sich dabei verschiedene Alternativen beim Design eines Netzes. Zunächst muß durch die Beantwortung folgender Fragestellungen die allgemeine Struktur des Netzwerkes festgelegt werden:

- Wieviele Elemente werden für das zu lösende Problem insgesamt benötigt?
- Sollen die Elemente des Netzes in einer Schicht oder mehreren angeordnet sein?
- Wieviele Verbindungen sollen zwischen den einzelnen Elementen bestehen, und wie sollen sie aufgebaut werden?

Fragen nach der Art und Anzahl der Elemente eines Netzwerkes sowie nach deren Verbindungen untereinander, werden durch die ausgewählte Netzwerkarchitektur bestimmt. Die für das ACEPS ausgewählte Netzwerkarchitektur wurde bereits in 4.2 vorgestellt und wird im folgenden noch detaillierter beschrieben werden.

Weitere Fragen betreffen die Lernphase und die Lern- bzw. Leistungsfähigkeit des Netzes.

- Welche Art von Aktivierungsfunktion soll benutzt werden?
- Wenn das Netz fähig ist, eine gewisse Aufgabenstellung zu lernen, wieviele Beispiele sind dann nötig, um eine zufriedenstellende Performanz auf den Testdaten zu garantieren?

- Wieviele Trainingsläufe müssen durchgeführt werden, um diese Leistungsfähigkeit zu erreichen?
- Kann das Netz in Realzeit lernen, d.h. während es seine Aufgabe erfüllt, oder muß die Trainingsphase von der Arbeitsphase getrennt sein?
- Ist das Netz in der Lage, von einer bekannten Aufgabe zu einer neuen noch nie dagewesenen zu verallgemeinern?

Designentscheidungen, die die Lernfähigkeit des ACEPS-Netzes durch Training betreffen, werden in Abschnitt 5.4 anhand der Beschreibung der Realisierung des Systems sowie den durchgeführten Tests und deren Ergebnissen (siehe Kapitel 6) näher erläutert.

5.3.2 Architektur des Netzwerkes

Wie schon erwähnt basiert die Architektur des ACEPS auf der in Abschnitt 4.2 vorgestellten Idee des adaptiven Erwerbs von Sprache. Grundlage des ACEPS ist ein neuronales Netzwerk, das aus zwei Schichten besteht: einer Eingabeschicht und einer Ausgabeschicht. Die Elemente der Eingabeschicht repräsentieren die Wörter des Sprachmodells des Netzwerkes. Ihre Anzahl kann nicht bei Initialisierung des Netzwerkes statisch vorgegeben werden, sondern vergrößert sich ständig je mehr Wissen sich das Netzwerk erwirbt, d.h. je öfter das Netzwerk trainiert und je größer sein Wortschatz wird. Die Elemente der Ausgabeschicht entsprechen den semantischen Aktionen, die auf den Eingabemustern ausgeführt werden sollen bzw. den Klassen, in die die Eingabemuster eingeordnet werden müssen. Die Menge der Ausgabeelemente wird durch den Benutzer bei Aufruf des Systems festgelegt.

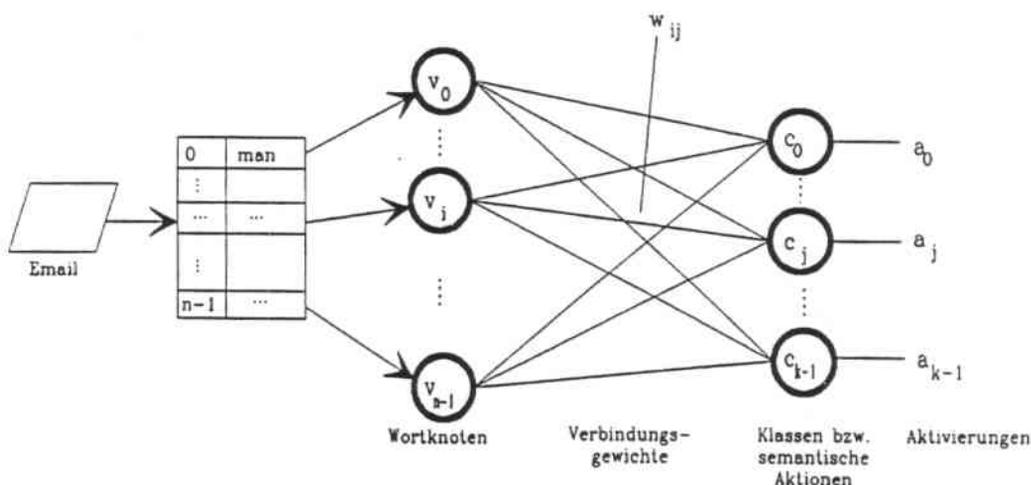


Abbildung 5.4: Das Netzwerk des ACEPS

Bild 5.4 stellt die Architektur des ACEPS-Netzwerkes dar. Die Verbindungen der Netzelemente sind vollständig, d.h. jedes Element der Eingabeschicht ist mit jedem Element

der Ausgabeschicht verbunden. Die Gewichte dieser Verbindungen werden durch den Trainingsalgorithmus des ACEPS bestimmt. Im Rahmen dieser Diplomarbeit wurden verschiedenen Varianten miteinander verglichen. Diese Algorithmen werden im nächsten Abschnitt näher erläutert, die Ergebnisse der Gegenüberstellungen finden sich in Kapitel 6.

Eingabe des Netzwerkes ist unbeschränkter geschriebener Text in Form von elektronischen Postnachrichten. Die Knoten der Eingabeschicht des Netzwerkes repräsentieren das bisher vom ACEPS angetroffene Vokabular. Dieses ist in einem Lexikon abgespeichert. Für jedes Wort des Eingabetextes wird zunächst überprüft, ob es bereits im Wortschatz des Systems vorhanden ist. Ist dies nicht der Fall, so wird das Lexikon erweitert und eine neue Netzeinheit für das unbekannte Wort erzeugt. Alle Wörter des Lexikons sind mit einer eindeutigen Identifikation versehen. Die Identifikationen werden den Wörtern in der Reihenfolge ihres zeitlichen Auftretens vergeben. So hat das Wort, das als erstes in der allerersten Email angetroffen wurde, die Identifikation 1. Das Lexikon ist nicht lexikalisch geordnet, sondern in einem Binär-Baum organisiert. Als Wurzel des Binär-Baumes wird das Wort *man* verwendet, welches die Identifikation 0 besitzt. Ein Wort, das noch nicht im Lexikon enthalten ist, erhält die Identifikation *id* (= Anzahl der Wörter im Lexikon + 1) und wird in den Binär-Baum eingefügt. Befindet sich das im Eingabetext der elektronischen Post vorkommende Wort im Lexikon, wird das Eingabeelement aktiviert, das dieses Wort verkörpert. Angesprochen werden die Eingabeknoten durch die den einzelnen Wörtern zugeordneten Identifikationen. Diese laufen, wie schon erwähnt, von 0 bis zur Anzahl der Wörter im Lexikon. Die Identifikation eines Wortes ergibt sich dann durch den alphabetischen Vergleich des Wortes mit den Elementen des Binär-Baumes. Sind alle im Lexikon vorhandenen Wörter des Eingabetextes ermittelt, werden abhängig von der Aktivierung der Eingabelemente die Ausgabelemente aktiviert. Schließlich wird die semantische Aktion ausgeführt, die, je nach Klassifikation der elektronischen Post, die maximale Aktivierung aufweist.

In einer weiteren Version des ACEPS wurde die Architektur des ihm zugrundeliegenden Netzwerkes durch eine Zwischenschicht ergänzt². Diese Schicht befindet sich zwischen Eingabe- und Ausgabeknoten des Netzes. In [Gor90] werden als Elemente einer solchen Zwischenschicht Phrasen bzw. Redewendungen vorgeschlagen. Aufgrund der Fülle möglicher Wortphrasen werden die Redewendungen schon in dieser Arbeit auf benachbarte Wortpaare begrenzt. Durch die unbeschränkte Eingabe in das ACEPS verfügt das System über einen riesigen Wortschatz und damit eine sehr große Menge von Eingabelementen³. Eine Aufnahme aller Wortpaarkombinationen der in der Eingabeschicht dargestellten Wörter würde zu einer recht unrealistischen Anzahl von Netzwerkelementen führen. Daher wird im hier vorliegenden Fall die Aufnahme von neuen Elementen durch eine weitere Beschränkung (siehe Abschnitt 6.2.4) eingegrenzt: Die Elemente der Zwischenschicht repräsentieren ausschließlich Wortpaare, die in den als Trainingsmails verwendeten Eingabetexten vorkommen. Man kann jedoch nicht direkt von einer verdeckten Schicht des Netzwerkes sprechen, sondern die Elemente der mittleren Schicht können als zusätzliche

²Die Veränderungen, die sich durch diese Umgestaltung in der Leistungsfähigkeit des ACEPS ergeben, können in Abschnitt 6.2.4 nachgelesen werden.

³Die genaue Anzahl läßt sich aus den Tabellen in Kapitel 6 entnehmen, die die Ergebnisse der durchgeführten Tests enthalten.

Eingabeelemente betrachtet werden. Die Architektur des sich ergebenden Netzwerkes nach Aufnahme von Wortpaaren ist in Bild 5.5 dargestellt. Die Aktivierung der Wort-

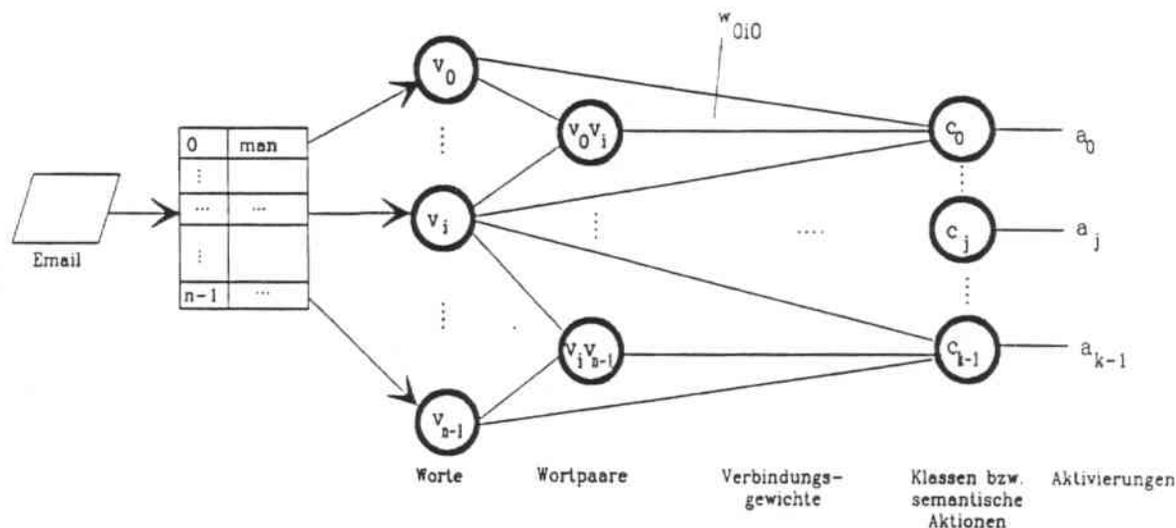


Abbildung 5.5: Das Netzwerk des ACEPS nach Ergänzung durch eine Zwischenschicht

paare hängt von der Aktivierung der Wörter ab, die dieses Wortpaar bilden.

Die Bestimmung der Gewichte der Verbindungen in beiden Netzarten erfolgt durch die im folgenden Abschnitt vorgestellten Trainingsalgorithmen.

5.3.3 Der Trainingsalgorithmus

Die Leistungsfähigkeit eines künstlichen neuronalen Netzwerkes wird vor allem durch seine Lernfähigkeit und die Länge des Lernvorgangs bestimmt. Lernen erfolgt mit Hilfe eines Trainingsalgorithmus, der angibt, in welchem Maße die Gewichte der Verbindungen zwischen den einzelnen Elementen verändert werden. In den mit dem ACEPS durchgeführten Experimenten wurde die Eingabe des Netzes vorverarbeitet, auf einen bestimmten Umfang gekürzt und teilweise durch Rauschen verzerrt. Die Architektur des Netzwerkes wurde durch verschiedene Strategien verändert: Hinzufügen neuer Schichten, Einfügen neuer Netzelemente oder Manipulation von Verbindungen. Neben diesen verschiedenartigen Ausführungsvarianten existieren außerdem noch mehrere Versionen von Trainingsalgorithmen, die in der Lernphase des ACEPS eingesetzt werden können. Die erste Version berücksichtigt vor allem den adaptiven Erwerb des Sprachmodells des ACEPS und ermöglicht ein sequentiell durchführbares, sehr rasches Lernen. Dieser adaptiven Trainingsprozedur wurde als zweite Variation ein herkömmlicher Backpropagation-Algorithmus in seiner einfachsten Form gegenübergestellt. Schließlich erwies sich die Verbindung beider Möglichkeiten als optimale Voraussetzung für gute Leistungen des ACEPS. Die dritte Art des Trainings besteht also aus einer Kombination des oben genannten adaptiven Trainingsalgorithmus mit den Prinzipien des althergebrachten Backpropagation-Algorithmus. Jede der drei Möglichkeiten wird in diesem Abschnitt beschrieben werden.

Die Version des simplen adaptiven Trainingsalgorithmus beruht wie die in Abschnitt 4.2 vorgestellte Theorie auf der Schätzung von Wahrscheinlichkeiten zur Bestimmung der Verbindungsgewichte. Während des Trainingsvorgangs werden alle Emails der Trainingsmenge nacheinander durchlaufen und die Wahrscheinlichkeiten $P(c_j)$ für $j \in \{1, \dots, k\}$ des Auftretens der Klassen bzw. semantischen Aktionen c_1, \dots, c_k bestimmt. Im Falle des ACEPS handelt es sich nicht um semantische Aktionen, sondern um verschiedene Fachbereiche, in die die Emails eingeordnet werden sollen. Das Ausgeben eines Etiketts für eine Email kann jedoch genauso durch eine beliebige semantische Aktion ersetzt werden. Ebenfalls wird für jedes Wort v_i für $i \in \{1, \dots, n\}$ (bei einer Anzahl von n Worten im Lexikon), das jemals in einer Email angetroffen wurde und somit zum Vokabular des Systems gehört, die bedingte Wahrscheinlichkeit $P(c_j|v_i)$ unter der Voraussetzung eines Wortes v_i bestimmt. Der Logarithmus der Division der bedingten Wahrscheinlichkeit $P(c_j|v_i)$ durch die Ausgabewahrscheinlichkeit $P(c_j)$ bestimmt das Gewicht der Verbindung des Wortes v_i zur Aktion c_j :

$$w_{ij} = \log \frac{P(c_j|v_i)}{P(c_j)} = \log \frac{P(c_j, v_i)}{P(c_j) \cdot P(v_i)} = \log \frac{C(c_j, v_i)}{C(P(c_j)) \cdot C(v_i)}$$

Das Lernen des Netzwerkes erfolgt so in einem sequentiellen Durchlauf durch alle bestehenden Verbindungen. Anders als bei einem Backpropagation-Algorithmus finden keinerlei Iterationen statt, es handelt sich um ein sogenanntes *one-shot-learning*. Dies wird durch den statistischen Ansatz der Gewichtsbestimmung ermöglicht. Gewichte können nur verändert werden, wenn neue Mails zum Trainingskorpus hinzukommen und sich somit die Wahrscheinlichkeitsverteilung des Vorkommens bestimmter Wörter ändert. Durch das Einstellen der Gewichte aller Verbindungen des Netzes in nur einem Durchgang benötigt der Lernvorgang also deutlich weniger Zeit als es die generalisierte Delta-Regel des Backpropagation-Algorithmus erfordern würde. Auch ist es nicht von Bedeutung, in welcher Reihenfolge die vorliegenden Muster dem Netzwerk zum Lernen dargeboten werden.

Die zweite Art und Weise, mit der innerhalb des ACEPS gelernt werden kann, wird durch einen Backpropagation-Algorithmus realisiert. Die detaillierte Beschreibung eines Backpropagation-Trainingsalgorithmus findet sich bereits in Kapitel 4. In einem Vorwärtsdurchlauf durch das Netzwerk werden die Aktivierungen aller Netzelemente durch eine Trainingsmail ermittelt und mit den erwünschten Ausgaben verglichen. In einem weiteren Durchgang wird in umgekehrter Richtung der Fehler jedes Elements berechnet und die Gewichte der Netzverbindungen neu angepaßt. Die Gewichte werden also sofort nach jedem Lernschritt verändert. Die Initialisierung der Gewichte vor der ersten Ausführung des Backpropagation-Algorithmus erfolgt durch Zufallswerte zwischen -0,3 und +0,3.

Letztendlich wurde eine Kombination der beiden gerade vorgestellten Lernmethoden als Trainingsprozedur verwendet. Die Anwendung des ersten Trainingsalgorithmus sorgt für ein rasches Lernen in nur einem Durchgang durch das gesamte Netzwerk des ACEPS. Die Überlegung war, dieses Vorwissen für eine Anwendung des Backpropagation-Algorithmus auszunutzen, und so den Lernvorgang zu beschleunigen. Tatsächlich ergab die Verbindung beider Algorithmen nicht nur eine Verkürzung der Trainingsdauer, sondern auch eine Verbesserung der Performanz. Die genauen Testresultate können in Kapitel 6 nachgelesen werden.

5.4 Realisierung

Ein lauffähiger Prototyp des ACEPS wurde in der Sprache C [Ker88] unter dem Betriebssystem Unix [Gul88] realisiert. Die Komponenten der in Abschnitt 5.1 dargestellten Systemarchitektur bilden die Grundlage der Modulstruktur des implementierten Systems. Im folgenden wird jedoch nicht auf implementierungstechnische Details eingegangen, sondern nur grob die in den Modulen realisierte Funktionalität umrissen.

- main** In diesem Modul werden die vom Benutzer eingegebenen Parameter in Aufrufe der einzelnen Komponenten umgesetzt. Die Abfolge aller Aktionen während eines Ablaufs des ACEPS werden festgelegt. Dazu gehört auch Art und Struktur des Trainings.
- preprocess** Hier findet die Vorverarbeitung einer Email statt, die unnötige Leitweginformation aus dem Kopf der Nachricht wird entfernt.
- test** In der Testkomponente des Systems wird der Aktivierungszustand des Netzes bei Eingabe einer Email berechnet und die Klassifikation dieser Eingabe ermittelt.
- train** Dieses Modul realisiert das Training über alle Muster der Trainingsmenge. Die Gewichte des Netzwerkes werden aufgrund der Berechnung von Wahrscheinlichkeiten verändert.
- backprop** Das Netz lernt durch Backpropagation-Training. Die Verbindungsgewichte des Netzwerkes werden durch die berechneten Fehler verändert.
- net** Alle Netzwerkstrukturen werden durch dieses Modul verändert. Aktionen auf dem Netzwerk des ACEPS können sein: Einfügen einer neuen Schicht, Einfügen eines neuen Elements, Schaffen neuer Verbindungen zwischen Elementen usw.
- lexicon** Innerhalb dieses Moduls werden alle Veränderungen der Strukturen im Lexikon realisiert. So werden neue Wörter in den Binärbaum eingefügt oder auch einfach nur Identifikationen bereits vorhandener Wörter ermittelt und an das Netzwerk weitergegeben.
- analysis** Die innerhalb der Klassifikation der Trainingsmenge gemachten Fehler werden analysiert. Mit Hilfe von *net* werden neue Knoten, die Wortpaare repräsentieren, ins Netz aufgenommen.
- read** Alle in Betriebssystem-Dateien abgelegten Strukturen des ACEPS werden eingelesen. Es handelt sich dabei um Lexikon, Netzwerk und die Liste nicht deskriptiver Wörter (LNW).
- save** Alle im Laufe einer Benutzung eventuell veränderten Strukturen wie Lexikon oder Netz des ACEPS werden in Betriebssystem-Dateien abgelegt.

Die bis dahin vorgestellten Module befassen sich mit der Grundstruktur des ACEPS. Durch die verschiedenen durchgeführten Testsimulationen wurden noch weitere Module geschaffen, die nur für einzelne Testfälle benötigt werden:

bignet Statt eines Netzes wird als Kern des ACEPS ein Netzwerk aufgebaut, das aus 3 Teilnetzen besteht.

morph Der Wortschatz des ACEPS wird durch verschiedene Restriktionen eingeschränkt.

con_learn Hier wird das Training des Netzes durch *constructive learning* vorgenommen.

smooth Dieses Modul sorgt für eine Glättung der Parameter des Netzes während des Trainings durch Cooccurrence-Smoothing.

5.5 Anwendungen des ACEPS

Die bisherigen Tests des ACEPS erfolgten durch eine einfache Benutzeroberfläche, da vor allem das Ziel im Vordergrund stand, die Leistungsfähigkeit des Systems mit allen zur Verfügung stehenden Mitteln zu steigern. Der endgültige Einsatz des Systems erfolgt jedoch unter Zuhilfenahme des Unix- und X-Window Tools *Xmh*. Das Programm *Xmh* bietet als graphischer Aufsatz auf den Mailer *mh* eine benutzerfreundliche Fenster-Oberfläche, in die das ACEPS integriert werden kann. Die Systemarchitektur des durch Kombination beider Anwendungen entstehenden Systems ist in Bild 5.6 dargestellt. Ein Unterschied zur simplen Version des ACEPS besteht darin, daß das ACEPS nicht mehr direkt vom Benutzer aufgerufen, sondern durch die *Xmh*-Oberfläche gestartet wird. Eine weitere Abweichung ist die Tatsache, daß eine elektronische Postnachricht nach Bearbeitung durch die Testkomponente des ACEPS durch eine Aufbereitungskomponente verändert wird.

Die Wirkungsweise des ACEPS innerhalb der vorgestellten Architektur ist dann die folgende: Nach Erzeugung eines individuellen ACEPS wird ankommende Mail nicht sofort in der *mailbox* des Benutzers abgelegt, sondern erst vorklassifiziert. Dieser Vorgang entspricht der bisher üblichen Testphase: die Email wird vorverarbeitet und durch die Testkomponente ausgewertet. Aufgrund der Aktivierung des Netzes des ACEPS wird eine Klassifikation der Email vorgenommen. Das Ergebnis dieser Klassifikation wird nun durch die Email-Aufbereitungskomponente mit in die Nachricht aufgenommen. Danach wird sie auf dem normalen Weg in die *mailbox* des Benutzers befördert, das ACEPS greift somit in den Übertragungsvorgang ein und leitet die Email weiter. Beim Lesen neuer Mails wird die Nachricht dann auf die übliche Weise in den Ordner „inbox“ eingliedert. Der Benutzer hat so die Möglichkeit, die Klassifikation durch ACEPS zu akzeptieren oder eine Korrektur vorzunehmen. Durch die Einordnung einer Mail in einen Ordner seiner Wahl wird ein vorklassifiziertes Ein-/Ausgabepaar erzeugt. Die automatische Etikettierung der Emails durch ihre Übertragung in verschiedene Verzeichnisse unter Unix erleichtert somit die Benutzung des ACEPS innerhalb der *Xmh*-Mail-Oberfläche. Ein Benutzer von *Xmh* bewahrt bereits gelesene Mails in bestimmten, seinen Prioritäten entsprechenden, Ordnern auf. Diese vorklassifizierten Nachrichten werden als Trainingsdaten zum Aufbau eines Netzwerkes nach seinen Wünschen verwendet.

Eine andere Möglichkeit wäre, eine Mail entsprechend ihrer Klassifikation durch das ACEPS in den entsprechenden Ordner zu transportieren anstatt sie aufbereitet in den üblichen Ordner „inbox“ aufzunehmen. Ein Benutzer könnte dann immer die Mails zuerst lesen, die sich in seinem bevorzugten Ordner befinden. Bevor das ACEPS jedoch

nicht mit nahezu 100%iger Sicherheit die Wünsche eines Benutzers ausführt, empfiehlt es sich, falsch klassifizierte Mail nicht in selten gelesenen Ordnern verschwinden zu lassen. Zudem muß in diesem Fall angezeigt werden, in welchen Ordnern sich ungelesene Mail befindet, da kein Benutzer alle in Frage kommenden Ordner nach neuer Post durchsuchen möchte. Bei einer Einteilung der Ordner nach Fachgebieten wie es bei vielen Menschen üblich ist, wäre alte bereits gelesene Post dann auch mit neuer ungelesener vermischt.

Ein solches Vorgehen ist zweifellos nicht praktikabel und der Aufnahme der Klassifikation in die Mail selbst weitaus unterlegen. Hinsichtlich dieser Methode ist als weitere Verfeinerung auch eine Auflistung aller ankommenden Post nicht nach Ankunftszeit, sondern Wichtigkeit denkbar. Dabei hat jede Mail eine Markierung für einen bestimmten Fachkorb, der der Benutzer entsprechen oder die er korrigieren kann. Trotzdem liest er in diesem Fall die für ihn mit höchster Priorität versehene Mail als erstes.

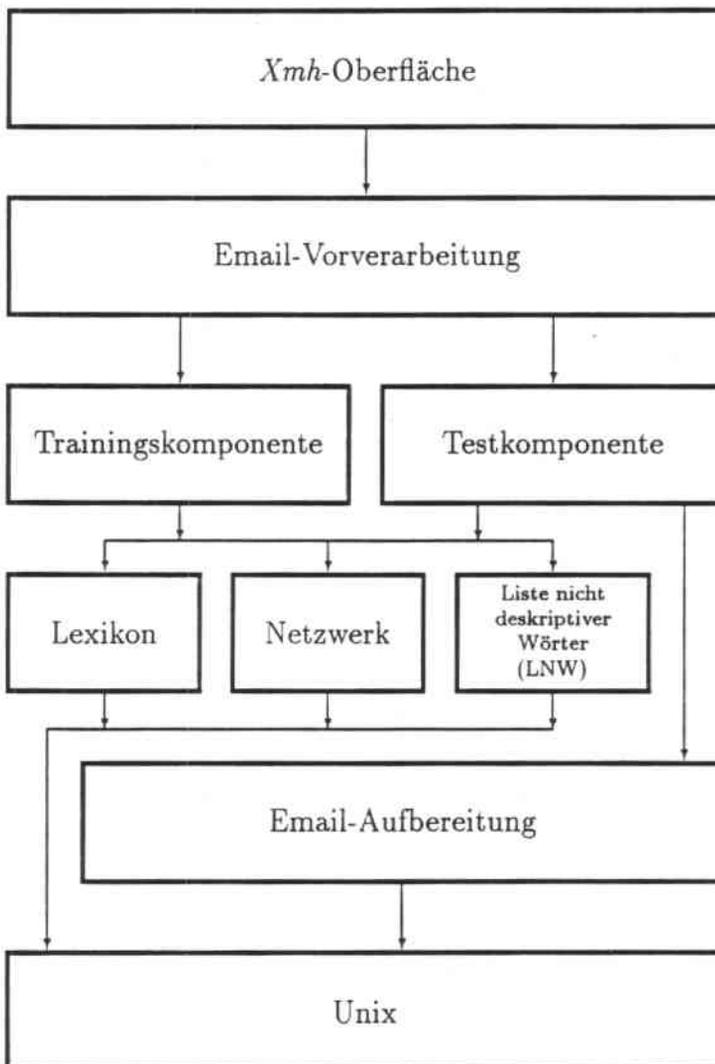


Abbildung 5.6: Systemarchitektur des ACEPS unter Einbeziehung des *Xmh*-Tools

Kapitel 6

Simulationen und Ergebnisse

Das folgende Kapitel beschreibt zunächst den experimentellen Aufbau und das experimentelle Vorgehen, das allen mit dem ACEPS durchgeführten Simulationen zugrunde liegt. Ein menschliches Vergleichsexperiment wird vorgestellt und der Wortschatz des ACEPS analysiert. Darauf folgen Beschreibungen der zum Testen des ACEPS verwendeten Verfahren und den sich daraus ergebenden Leistungen. Eine Zusammenfassung der wichtigsten Ergebnisse und eine Fehleranalyse der Test- und Trainingsperformanz schließen das Kapitel ab.

6.1 Experimentelles Vorgehen

Im Rahmen dieser Diplomarbeit standen mehrere DEC-Stations 5000/125 zur Verfügung. Auf dieser Testumgebung wurden sämtliche Simulationen durchgeführt.

Trainiert und getestet wurde das ACEPS mit einem Trainingskorpus von 946 Mails und einem Testkorpus von 258. Diese Mails stammen aus dem Archiv der Mailing-Liste „connectionists@cs.cmu.edu“. Die Mailing-Liste wurde von der Carnegie Mellon University in Pittsburgh, PA initiiert und beschäftigt sich mit konnektionistischen Themen. Sämtliche 1204 Emails sind in englischer Sprache verfaßt. Die elektronischen Postnachrichten wurden unverändert als Eingabe des ACEPS benutzt. Lediglich der Kopf der Nachrichten wurde bis auf Gegenstand der Email („subject“-Zeile)¹ und Absender („From“-Zeile) automatisch entfernt.

Die vom Netz des ACEPS zu lernenden bzw. zu klassifizierenden Nachrichten stehen alle mit Forschung und Praxis neuronaler Netzwerke in Zusammenhang. Es handelt sich dabei um Ankündigungen und *call for papers* von Konferenzen und sogenannten *Workshops*. Gegenstand dieser Konferenzen und *Workshops* sind neuronale Netzwerke bzw. Anwendungen wie Signal- und Sprachverarbeitung, die häufig mittels konnektionistischer Ansätze realisiert werden. Verbreitet werden auch Stellenangebote an Universitäten oder Forschungsinstituten und der Industrie, die sich mit Neurobiologie oder neuronalen Netzen im allgemeinen befassen. Der Abonnent von „connectionists@cs.cmu.edu“ erfährt von

¹Stichworte, die den Gegenstand der Nachricht zusammenfassen.

Vorankündigungen gerade veröffentlichter Artikel, die via *ftp* (*file transfer program*) interessierten Lesern zugänglich sind sowie Literaturlisten zu bestimmten Problemstellungen. Ebenso trifft man auf Bekanntmachungen neuer Netzwerksimulatoren, die als *public domain* den Benutzern zur Verfügung stehen. Nicht zuletzt findet in dieser Mailing-Liste ein reger Gedankenaustausch zwischen neuronalen Netzwerkforschern über die verschiedensten Themen statt. Diese fachlichen Diskussionen stellen den größten Anteil der ankommenden Nachrichten dar. „Connectionists@cs.cmu.edu“ wird damit zu einem Kommunikationszentrum für Neurobiologen, Physiker, Informatiker, Psychologen und Linguisten. Bisweilen findet der „konnektionistische“ Leser natürlich auch Mail in seinem Briefkasten, die eigentlich nicht in eine solche Mailing-Liste gehört. Dazu zählen die Bitten, auf die Liste der Abonnenten der Liste gesetzt zu werden bzw. die Aufforderung, wieder von der Liste gestrichen zu werden, sowie auch Nachrichten, die versehentlich an die Adresse „connectionists@cs.cmu.edu“ geschickt wurden. Unter Berücksichtigung dieser Vielfalt verschiedener Nachrichten wurden fünf Themengebiete ausgewählt, in die die 1204 Mails vom ACEPS eingeordnet werden sollen:

- Konferenzankündigungen
- Stellenangebote
- Veröffentlichungen
- Fachliche Diskussion
- Unwichtiges

Dabei handelt es sich um subjektiv von einem potentiellen Benutzer nach seinen Interessen und Schwerpunkten ausgewählte Cluster.

Die 1204 Mails wurden von Hand klassifiziert. Statt sie explizit mit einem Etikett zu versehen, wurden sie in fünf verschiedene Unterverzeichnisse kopiert. Die Klasse einer Email läßt sich dann aufgrund ihres Pfadnamens bestimmen.

Die Auswahl der 1204 Nachrichten aus dem Archiv der Mailing-Liste erfolgte willkürlich. Aus den Emails eines bestimmten Zeitraums wurde eine repräsentative Selektion getroffen, indem versucht wurde, einen Kompromiß zwischen einer Gleichverteilung aller fünf Gruppen innerhalb dieser Menge und einer der Wirklichkeit der Mailing-Liste entsprechenden Verteilung zu finden. Bei vier der fünf vorgegebenen Klassen waren genügend Mails vorhanden, um eine Gleichverteilung zu erreichen. Lediglich Stellenangebote waren so unterrepräsentiert, daß ihr Anteil nicht dem der anderen Klassen entspricht. So entstand eine Verteilung von je 280 Mails aus den Klassen „Konferenzankündigungen“, „Veröffentlichungen“, „Fachliche Diskussion“ und „Unwichtiges“ und 84 Mails der Klasse „Stellenangebote“. Innerhalb dieser Umgebung wurden alle Tests mit verschiedenen Parametern ausgeführt.

Tabelle 6.1 gibt noch einmal die wichtigsten Kenndaten des verwendeten Test- und Trainingskorpus wieder. Darin wird die Anzahl der vorhandenen Test- und Trainingsmails pro Themengebiet angegeben. Es folgt die durchschnittliche Anzahl von Zeilen, aus der eine Email besteht. Darüber hinaus enthält die Tabelle Angaben über die Gesamtzahl der in

Themen- gebiet	Anzahl Trainingsmails	Anzahl Testmails	Durchschnittliche Länge einer Email		Größe des Vokabulars einer Email (in Worten)
			in Zeilen	in Worten	
Konferenz- ankündigungen-	220	60	130	601	340
Fachliche Diskussion	220	60	58	324	190
Stellen- angebote	66	18	51	268	172
Unwichtiges	220	60	21	86	61
Veröffentlichungen	220	60	72	349	221

Tabelle 6.1: Kenndaten der Test- und Trainingsmails

einer Email vorkommenden Wörter sowie die Menge voneinander verschiedener Wörter, was der Größe des Vokabulars einer Nachricht entspricht.

6.1.1 Vorgehen beim Training

Zum Training des ACEPS wurden zwei verschiedene Methoden benutzt. Bei der ersten Methode ist die Vorgehensweise innerhalb einer Simulation die folgende: die 946 Mails des Trainingskorpus werden in Gruppen zu je 86 Mails aufgeteilt. Das Netzwerk wird in 11 Zyklen mit je 86 Mails trainiert. Nach jedem Zyklus erfolgt eine Bewertung der Leistung durch die 258 Mails große Testmenge. Die Einteilung in 11 Zyklen findet statt, um zu überprüfen, ob eine Leistungssteigerung zu verzeichnen ist, je öfter das ACEPS trainiert wird. Das Training besteht somit aus 11 Lern- und 11 Ausführungsphasen, die miteinander verglichen werden können.

Die zweite Methode orientiert sich am herkömmlichen Vorgehen eines Backpropagation-Algorithmus. Alle 946 Trainingsmails werden in Stapelverarbeitung eingelesen und das Netz wird innerhalb einer einzelnen Lernphase mit der gesamten Menge vorklassifizierter Beispiele trainiert. Auf die Lernphase folgt eine einzelne Ausführungsphase, anhand deren Ergebnis die tatsächliche Leistungsfähigkeit des Netzes bewertet wird.

6.1.2 Vorgehen beim Testen

Die Leistungsfähigkeit des ACEPS wurde, wie schon im vorigen Abschnitt beschrieben, nach jeder Trainingsphase durch die Menge der Testmails beurteilt. Daraus ergeben sich bei der ersten Trainingsmethode 11 Testergebnisse, die miteinander verglichen werden können, da sie auf der gleichen Beurteilungsgrundlage basieren. Aufgrund der Art des angestrebten Einsatzes des ACEPS (siehe auch Abschnitt 5.5) wird später jede einmal

vom ACEPS bearbeitete Nachricht als Trainingsmuster dienen, um eine immer bessere Performanz des Systems zu gewährleisten. Dieser Ablauf wird auch jetzt schon beim Training des ACEPS berücksichtigt. Emails, die in der Menge der Trainingsmuster aufgenommen werden, werden nicht einfach wie in Abschnitt 5.2 beschrieben in die Beispielmengemenge aufgenommen und das Netz und Lexikon erweitert, sondern durchlaufen vorher einen Testzyklus, in dem sie wie Emails aus der Testmenge behandelt werden. Die zeitliche Abfolge zwischen Lern- und Ausführungsphase kann unterschiedlich gewählt werden. Theoretisch ist es möglich, jede Mail als Testmail an das ACEPS anzulegen, sie dann als Trainingsmuster ins System aufzunehmen und das Netz mit dem neuen Wissen zu trainieren. Damit ist aber keine Performanzmessung über eine ganze Reihe Mails möglich, da jede neue Mail durch ein anderes, besseres ACEPS bearbeitet werden würde als die vorhergehende. Um also, außer der „offiziellen“ Menge von Testmails die Performanz des Netzwerkes auch anhand einer beliebigen Menge von Emails überprüfen zu können, werden auch die Trainingsmails „vor“getestet. Jede Portion von Trainingsmustern wird erst als Quasi-Testmenge an das Netz angelegt und klassifiziert. Aus dieser Klassifikation wird eine Genauigkeitsrate ermittelt und erst dann wird das neu hinzugekommene Wissen benutzt, um das Netzwerk nochmals zu trainieren. Aus dieser Art des Testens, die im folgenden „Zyklustest“ genannt werden soll, entstanden weitere 11 Zyklusergebnisse. Diese Ergebnisse lassen sich zwar nicht direkt miteinander vergleichen, da es sich bei jedem Testzyklus um eine andere Menge von Nachrichten handelt. Man erhält jedoch Ergebnisse recht unterschiedlicher Testmengen, die Aussagen über die Verwendbarkeit und Leistung des ACEPS erlauben.

Von Interesse ist sicherlich auch, wie gut ein Netzwerk seine Trainingsmuster gelernt hat. Daher findet nach jedem Zyklus ein Test der Trainingsmenge des Netzes statt. Dieser Test erlaubt eine Analyse der Stabilität einmal gelernter Muster.

6.1.3 Klassifikation durch Menschen

Zweck dieser Arbeit ist es, eine möglichst hohe Genauigkeit bei der Klassifikation elektronischer Postnachrichten zu erhalten. Dies geschieht mit Hilfe eines künstlichen neuronalen Netzes. Ohne dabei zu sehr auf Ähnlichkeiten zwischen den Elementen eines neuronalen Netzwerkes und den Neuronen des Hirns einzugehen, ist ein erstrebenswertes Ziel der konnektionistischen Forschung die Nachbildung menschlicher Fähigkeiten und menschlichen Wissens. So soll auch in dieser Forschungsarbeit oberste Priorität sein, Emails so zu klassifizieren wie es der Mensch als Benutzer des ACEPS tun würde. Sicherlich ist eine Klassifikation von Emails der Mailing-Liste „connectionists@cs.cmu.edu“ keine objektive Angelegenheit, bei der es nur eine absolute und richtige Klassifikation gibt. Gerade das hier behandelte Anwendungsgebiet ist von hoher Subjektivität und dem persönlichen Empfinden eines Menschen geprägt. Nicht nur ist die Beurteilung der Klassenzugehörigkeit einer Email von Mensch zu Mensch verschieden, sondern auch die Kriterien eines einzelnen Menschen können in der Zeit variieren und im Laufe der Zeit können sich die Grenzen zwischen verschiedenen Klassen verändern.

Aus diesem Grund wurde in dieser Arbeit sehr viel Wert auf ein flexibles ACEPS gelegt, das sich leicht und vor allem in tolerabler Zeit auf veränderte Benutzerwünsche einstellen kann. Die erste in Abschnitt 5.3.3 vorgestellte adaptive Trainingsprozedur erfüllt

diese Forderung. Sie ermöglicht ein rasches Lernen des dem ACEPS zugrundeliegenden neuronalen Netzes und paßt sich innerhalb kürzester Zeit einer neuen Aufgabenstellung an. Selbst bei einer nur sehr kleinen Menge von durch den Menschen vorklassifizierten Trainingsbeispielen erreicht das ACEPS eine gute Performanz. Um die Qualität der Genauigkeit des ACEPS besser beurteilen zu können, wurde ein recht ungewöhnliches aber doch interessantes und nachvollziehbares Experiment durchgeführt. Vier menschliche Testpersonen wurden in einem Versuch (MENSCH) gebeten, die Menge der Testmails in die vorgegebenen Klassen einzuordnen.

Die absolut vorgegebene Klasseneinteilung durch eine einzelne Person, die die Daten etikettiert hatte, unterscheidet sich dabei zweifellos von der Klasseneinteilung, die ein anderer Mensch vorgenommen hätte. Ebenso ist, selbst bei vorgegebenen Klassen, keine 100%ige Übereinstimmung zweier Menschen zu erreichen. Sicherlich soll hier keineswegs außer Acht gelassen werden, daß die Benutzung des ACEPS für einen Menschen nur dann sinnvoll ist, wenn es ihm seine Arbeit erleichtert und dies auch mit nahezu 100%iger Genauigkeit tut. Trotzdem ist die Beurteilung der Leistungsfähigkeit des Netzwerkes anhand der Klassifikation verschiedener Menschen auch aufschlußreich. Die bei diesem Versuch erzielten Resultate liegen zwischen 71,3% und 83,3%, was einer durchschnittlichen Übereinstimmung von 79,4% entspricht. Die genauen Ergebnisse sind in Tabelle 6.2 zu finden. Das Netzwerk des ACEPS erreichte als bestes Ergebnis eine Performanz von 79,1%, die Ergebnisse sind also vergleichbar.

Beschreibung	Testperson	Parameter	Testleistung
MENSCH	1	keine	81,0%
	2	keine	81,8%
	3	keine	71,3%
	4	keine	83,3%
MENSCH_ANW	3	mit Anweisungen	82,2%
	4	mit Anweisungen	84,1%
MENSCH_ANF50	5	Einschränkung der Emails auf die ersten 50 Worte	81,4%

Tabelle 6.2: Menschliche Testergebnisse

Eine Erwartung an das ACEPS ist, dieselbe oder bessere Leistung als die der Testpersonen zu erbringen — hat es doch auf, ebenfalls vom Erstbenutzer vorgegebenen, subjektiv eingeordneten Trainingsmails gelernt. Diese Möglichkeit war den Testpersonen ja verwehrt geblieben. Sie hatten keinerlei Wissen über die subjektiven Kriterien des Erstbenutzers, das Netz war also eigentlich bevorzugt, da es die Möglichkeit gehabt hatte, bestimmte repräsentative Merkmale der verschiedenen Klassen zu „lernen“. Den Testpersonen war der Zugang zu diesem Vorwissen völlig versperrt geblieben.

Um daher eine vergleichbare Messung menschlicher Performanz durchzuführen, wurde

noch ein Test vorgenommen, in dem zwei Testpersonen die Testmails mit einem Katalog von Anweisungen erneut bearbeiteten (MENSCH_ANW). Die Anweisungen bestanden aus einer Zusammenstellung von Regeln, die der Denkweise der Person entsprachen, die die Erstklassifikation vorgenommen hatte:

1. Zu der Klasse der „Konferenzankündigungen“ zählen:
 - Konferenz- und *Workshop*-Ankündigungen
 - Vorträge bzw. Seminare an Universitäten
2. In „Fachliche Diskussion“ sollen eingeordnet werden:
 - Diskussionen über bestimmte Fragen zwischen Mitgliedern der Mailing-Liste
 - Bitten um allgemeine Information und Verweise auf Literaturstellen zu speziellen neuronalen Netzwerk-Problemen
3. Der Ordner „Stellenangebote“ beinhaltet:
 - Stellenangebote von Universitäten und Firmen
 - *summer internships* oder *summer jobs* in Firmen bzw. *graduate courses* an Universitäten
 - Angebote von *postdoctoral training* und *PhD*-Kursen
4. Zu „Unwichtiges“ zählen:
 - Bitten um Herausnahme aus oder Hinzufügen zur Mailing-Liste
5. Die Klasse der „Veröffentlichungen“ soll enthalten:
 - Ankündigungen neuer Veröffentlichungen
 - Ankündigungen über die Vergabe von Preisen und Auszeichnungen für die „beste Veröffentlichung“
 - Bekanntmachungen über zu erscheinende *proceedings* von Konferenzen
 - Mitteilungen über Sonderausgaben von Zeitschriften zu speziellen Themen
 - Netzwerk-Bibliographien, die über *ftp* abrufbar sind

Diese Regeln sollen einen Klassifikationsrahmen bieten, an dem sich die Testpersonen orientieren können, ohne allzu restriktiv zu sein. Zur Ausführung dieses Experiments wurden die Personen mit dem bisher besten und schlechtesten Ergebnis ausgewählt. Die Ergebnisse der beiden Testpersonen waren 82,2% und 84,1%. Bei beiden Probanden ist also eine Verbesserung der Klassifikationsleistung eingetreten. Bei Testperson 4 ist diese Verbesserung jedoch so geringfügig, daß zu vermuten ist, daß ihre Klassifikationsregeln auch schon beim ersten Test weitgehend denen der Person entsprachen, die die Vorklassifikation vorgenommen hatte. Die Resultate und die gemachten Fehler zeigen jedoch auch, daß die vorgegebenen Regeln immer noch genügend Freiheitsgrade erlauben, um ein eigenes, von der Vorklassifikation abweichendes, Klassifikationsschema aufzustellen.

Anhand eigener Erfahrung liegt die Vermutung nahe, daß ein Mensch eine Mail nicht vollständig liest, sondern nur den Anfang überfliegt. Diese Vermutung, verbunden mit der höheren Genauigkeit unseres Netzes, hat schon dazu geführt, dem ACEPS nur die ersten 50 Wörter einer Mail zur Verfügung zu stellen. Dasselbe Experiment (MENSCH_ANF50) wurde nun auch mit einer einzelnen menschlichen Testperson durchgeführt und erreicht eine Genauigkeit von 81,4%. Bei einer Einordnung in Fachgebiete wie sie das ACEPS vornimmt, scheinen also auch bereits die Anfänge einer Email auszureichen, um eine Klassifikation vorzunehmen.

Bei der Auswertung der Testergebnisse der menschlichen Probanden stellte sich heraus, daß jede Testperson eine Art Lieblingsordner hatte. In diesen wurden Nachrichten eingeordnet, die nach Meinung der Testperson nicht eindeutig in einen der vier anderen Ordner paßten. Diese Tatsache ergibt sich durch Auswertung der Konfusionsmatrizen der Testpersonen. Die Tabellen 6.3, 6.4, 6.5 und 6.6 zeigen die aufgestellten Konfusionsmatrizen. Sie werden nur für die ersten vier Versuche ohne Parameter aufgestellt, da die anderen Tests keine wesentlich veränderten Ergebnisse aufweisen.

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	46	0	1	1	3
Fachliche Diskussion	13	58	3	9	16
Stellen- angebote	0	0	14	0	0
Unwichtiges	0	1	0	50	0
Veröffentlichungen	1	1	0	0	41

Tabelle 6.3: Konfusionsmatrix Testperson 1

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	52	0	4	0	0
Fachliche Diskussion	0	35	0	2	1
Stellen- angebote	0	0	13	0	0
Unwichtiges	7	18	1	58	6
Veröffentlichungen	1	7	0	0	53

Tabelle 6.4: Konfusionsmatrix Testperson 2

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	29	0	3	0	1
Fachliche Diskussion	6	39	1	3	3
Stellen- angebote	3	0	13	0	0
Unwichtiges	5	10	0	52	5
Veröffentlichungen	17	11	1	5	51

Tabelle 6.5: Konfusionsmatrix Testperson 3

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	51	2	4	0	2
Fachliche Diskussion	2	43	1	3	3
Stellen- angebote	3	0	13	0	0
Unwichtiges	0	7	0	54	1
Veröffentlichungen	4	8	0	3	54

Tabelle 6.6: Konfusionsmatrix Testperson 4

Die Spalten geben die Etiketle der klassifizierten Emails an (Soll-Wert), die Zeilen entsprechen den tatsächlich von den Testpersonen vorgenommenen Klassifikationen (Ist-Wert). Aus den Tabellen ist ersichtlich, daß z.B. der bevorzugte Ordner von Testperson 1 „Fachliche Diskussion“, der von Testperson 2 „Unwichtiges“ war. Tabelle 6.3 zeigt, daß Testperson 1 die meisten Fehlklassifikationen in den Fachgebieten „Konferenzankündigungen“ und „Veröffentlichungen“ vornahm. In beiden Fällen wurden die Mehrheit der falsch klassifizierten Mails in den bevorzugten Ordner „Fachliche Diskussion“ eingeordnet. In den Ordner „Stellenangebote“ wurden nur Mails eingeordnet, die auch für diesen Ordner bestimmt waren. Dieser Bereich scheint sich besser als alle anderen von den übrigen Klassen abgrenzen zu lassen. In den verbleibenden Fachgebieten wurden zwischen einer und fünf Mails falsch klassifiziert. In diesen drei Klassen scheint eine objektive Abgrenzung voneinander schwieriger zu sein. Ähnliche Resultate lassen sich aus allen vier Konfusionsmatrizen ablesen.

Als Zusammenfassung wird daher eine Konfusionsmatrix (siehe Tabelle 6.7) erstellt, die alle vier Testpersonen berücksichtigt und den Durchschnittswert aller bisher aufgestellten Konfusionsmatrizen bildet. So wurden von den vier Testpersonen durchschnittlich 44,5

der bei der Vorklassifikation als „Konferenzankündigungen“ etikettierten Emails auch tatsächlich in dieses Themengebiet eingeordnet. In der Klasse „Unwichtiges“ waren es sogar 53,5 Testmails, die richtig klassifiziert wurden. Die restlichen 6,5 Emails wurden in falsche Themengebiete eingeordnet: Durchschnittlich 0,25 in „Konferenzankündigungen“, 4,25 in „Fachliche Diskussion“ und 2 der Mails in „Veröffentlichungen“. Keine der von der Vorklassifikation als „Unwichtiges“ eingestuften Emails wurde jedoch in den Ordner „Stellenangebote“ eingeordnet. Dies gilt auch für die Themengebiete „Fachliche Diskussion“ und „Veröffentlichungen“. Wie schon oben erwähnt, ist der Bereich „Stellenangebote“ klarer von den übrigen Klassen abgrenzbar als die übrigen Themengebiete.

	Konferenz-ankündigungen	Fachliche Diskussion	Stellen-angebote	Unwichtiges	Veröffent-lichungen
Konferenz-ankündigungen	44,5	0,5	3	0,25	1,5
Fachliche Diskussion	5,25	43,75	1,25	4,25	5,75
Stellen-angebote	1,5	0	13,25	0	0
Unwichtiges	3	9	0,25	53,5	3
Veröffentlichungen	5,75	6,75	0,25	2	49,75

Tabelle 6.7: Konfusionsmatrix aller Testpersonen im Durchschnitt

In einer idealen Konfusionsmatrix wären nur die Elemente der Hauptdiagonalen von Null verschieden, alle anderen Elemente gleich Null. Tabelle 6.8 zeigt die Konfusionsmatrix der Vorklassifikation, die die ideale Matrix darstellt.

	Konferenz-ankündigungen	Fachliche Diskussion	Stellen-angebote	Unwichtiges	Veröffent-lichungen
Konferenz-ankündigungen	60	0	0	0	0
Fachliche Diskussion	0	60	0	0	0
Stellen-angebote	0	0	18	0	0
Unwichtiges	0	0	0	60	0
Veröffentlichungen	0	0	0	0	60

Tabelle 6.8: Konfusionsmatrix der Vorklassifikation

Die Übereinstimmung aller Testpersonen mit der idealen Klassifikation wird durch die in Tabelle 6.9 wiedergegebene Matrix dargestellt. Sie veranschaulicht den durchschnittlichen Konsens der vier Testpersonen mit der Vorklassifikation in Prozent. Die Matrix ergibt sich aus der idealen Konfusionsmatrix in Tabelle 6.8 und der Konfusionsmatrix in

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	74%	- 1%	- 17%	- 1%	- 3%
Fachliche Diskussion	- 9%	73%	- 7%	- 7%	- 10%
Stellen- angebote	- 3%	- 0%	74%	- 0%	- 0%
Unwichtiges	- 5%	- 15%	- 1%	89%	- 5%
Veröffentlichungen	- 10%	- 11%	- 1%	- 3%	83%

Tabelle 6.9: Übereinstimmung der Testpersonen mit der Vorklassifikation

Tabelle 6.7, die die durchschnittliche Anzahl der von den Testpersonen in die verschiedenen Themengebiete eingeordneten Testmails zeigt. Die Elemente der Hauptdiagonalen in Tabelle 6.9 ergeben sich durch Division des entsprechenden Elements der Matrix der durchschnittlichen Klassifikation aller Testpersonen durch die ideale Einordnung der Vorklassifikation. Die übrigen Elemente der Matrix entstehen ebenfalls durch Division durch die ideale Einordnung. Da es sich dabei jedoch um Fehlklassifikationen handelt, wird statt der Übereinstimmung zur Vorklassifikation eine Fehlerrate angegeben. Die durchschnittlichen Fehlklassifikationen werden durch negative Prozentzahlen ausgedrückt. Dies resultiert in den in Tabelle 6.9 aufgeführten Prozentzahlen. 100%ige Übereinstimmung bzw. eine Fehlerrate von - 0% findet sich lediglich in den Themengebieten „Fachliche Diskussion“, „Unwichtiges“ und „Veröffentlichungen“. Aus diesen Kategorien wurde keine Email falsch in die Klasse der „Stellenangebote“ eingeordnet.

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	85%	- 3%	- 22%	- 0%	- 3%
Fachliche Diskussion	- 3%	72%	- 6%	- 5%	- 5%
Stellen- angebote	- 5%	- 0%	72%	- 0%	- 0%
Unwichtiges	- 0%	- 12%	- 9%	90%	- 2%
Veröffentlichungen	- 7%	- 13%	- 0%	- 5%	90%

Tabelle 6.10: Übereinstimmung der Testperson 4 mit der Vorklassifikation

Berechnet man den Kontingenzkoeffizienten, der den Zusammenhang zwischen Soll- und Ist-Kategorien der Testmails darstellt, ergibt sich der Wert 0,83. Dies zeigt, daß trotz einiger abweichender Klassifikationen ein hoher Zusammenhang zwischen der Klassifikation der vier Testpersonen und der Vorklassifikation besteht.

Zum Vergleich wird hier auch noch einmal der Konsens von Testperson 4, die die höchste Übereinstimmung mit der Vorklassifikation hatte, und der idealen Klassifikation aufgestellt. Tabelle 6.10 gibt die Ergebnisse wieder.

Schließlich wird nicht nur die Übereinstimmung der Testpersonen zur Vorklassifikation, sondern auch ihre Übereinstimmung untereinander ausgewertet. Nicht beachtet werden dabei Emails, die nur einmal von einer einzelnen Testperson anders klassifiziert worden waren als es dem Etikett der Nachricht entspricht. Es ergibt sich, daß insgesamt 51 Emails von mehr als zwei Testpersonen falsch klassifiziert wurden. Dabei wurden 7 von ihnen von allen vier Testpersonen falsch eingeordnet, 18 Stück von drei Testpersonen und 26 Emails von zwei Probanden. Eine weitere interessante Beobachtung in diesem Zusammenhang ist die Tatsache, daß zwei der 258 Dateien sogar von allen vier Testpersonen einheitlich „falsch“ klassifiziert wurden. Die vier Testpersonen ordneten die zwei Emails übereinstimmend in einen anderen Ordner ein als es der Vorklassifikation entsprach. Das beweist einmal mehr wie subjektiv eine solche Klassifikation sein kann. Fünf weitere Emails wurden von allen Personen in andere Ordner als es der Markierung entsprach, jedoch nicht in denselben eingeordnet.

Als Beispiel für die Übereinstimmung zwischen den Testpersonen werden zwei Matrizen aufgestellt. Tabelle 6.11 zeigt den Konsens der ersten beiden Testpersonen. Bei Berechnung dieser Werte wird nicht die Vorklassifikation als Absolutwert berücksichtigt, sondern lediglich die relative Übereinstimmung der Ist-Einordnung von Emails einer bestimmten Soll-Klasse.

	Konferenz-ankündigungen	Fachliche Diskussion	Stellenangebote	Unwichtiges	Veröffentlichungen
Konferenz-ankündigungen	88%	100%	25%	50%	25%
Fachliche Diskussion	7%	60%	25%	22%	6%
Stellenangebote	100%	100%	87%	100%	100%
Unwichtiges	13%	6%	5%	86%	14%
Veröffentlichungen	100%	14%	100%	100%	77%

Tabelle 6.11: Übereinstimmung der Testpersonen 1 und 2

Die dritte Zeile der Matrix veranschaulicht deutlich die Übereinstimmung beider Probanden bei der Einordnung der Stellenangebote. Die übrigen Klassen zeigen teilweise recht unterschiedliche Einordnungen. Testpersonen 3 und 4, dargestellt in Tabelle 6.12 weisen eine noch größere Übereinstimmung auf. Auch hier ragt die Klasse der „Stellenangebote“ aus den übrigen Klassen heraus. Die Einigkeit der zwei Testpersonen ist jedoch auch im Ordner „Fachliche Diskussion“ hoch.

Analysiert man die prozentuale Verteilung aller Emails in die verschiedenen Themengebiete ergibt sich ebenfalls eine interessante Beobachtung. Bei vier Testpersonen und je 258

	Konferenz- ankündigungen	Fachliche Diskussion	Stellen- angebote	Unwichtiges	Veröffent- lichungen
Konferenz- ankündigungen	56%	33%	75%	100%	5%
Fachliche Diskussion	33%	90%	10%	10%	10%
Stellen- angebote	100%	100%	100%	100%	100%
Unwichtiges	17%	70%	100%	96%	20%
Veröffentlichungen	24%	73%	50%	60%	96%

Tabelle 6.12: Übereinstimmung der Testpersonen 3 und 4

zu klassifizierenden Nachrichten ergibt sich eine Gesamtzahl von 1032 Emails. Von diesen 1032 Nachrichten entfallen je 23,3% in die Klassen „Konferenzankündigungen“, „Fachliche Diskussion“, „Unwichtiges“ und „Veröffentlichungen“ und 7% in „Stellenangebote“. Nach Bearbeitung durch die Testpersonen wurden von den insgesamt 1032 Nachrichten 19,3% in „Konferenzankündigungen“, 23,4% in „Fachliche Diskussion“, 5,7% in „Stellenangebote“, 26,6% in „Unwichtiges“ und 25% in „Veröffentlichungen“ eingeordnet. Besonders viele Nachrichten wurden also in den Ordner „Unwichtiges“ einsortiert. Betrachtet man weiter die Fehlklassifikationsrate pro Kategorie ragt diese Klasse mit 89% richtig klassifizierten Emails aus den anderen heraus, wo durchschnittlich fast 25% der Nachrichten falsch klassifiziert wurden. Tabelle 6.13 stellt diese Ergebnisse noch einmal detailliert dar.

Themen- gebiet	Anteil Soll-Klassifikation	Anteil Ist-Klassifikation	Richtig Klassifiziert	Falsch Klassifiziert
Konferenz- ankündigungen	23,3%	19,3%	74,2%	25,8%
Fachliche Diskussion	23,3%	23,4%	72,9%	27,1%
Stellen- angebote	7%	5,7%	73,6%	26,4%
Unwichtiges	23,3%	26,6%	89,2%	10,8%
Veröffentlichungen	23,3%	25%	82,9%	17,1%

Tabelle 6.13: Verteilung der Nachrichten pro Themengebiet

Abschließend soll auch die Kritik erwähnt werden, die von den Testpersonen während des Experiments geäußert wurde. Die vorgenommene Einteilung in 5 Fachbereiche wurde als zu eng bzw. uneindeutig empfunden. Einzelne Testpersonen wünschten sich zusätzliche Ordner oder hätten selbst eine ganz andere Einteilung vorgenommen. Dadurch erklärt sich die Unsicherheit vieler Einordnungen, die auf fehlender Eindeutigkeit der Klassen beruht, und einen hohen Prozentsatz von Fehlern verursacht.

6.1.4 Beschränkung des Wortschatzes

Wie schon in Kapitel 4.2 erwähnt, besteht das Netzwerk des ACEPS aus zwei Schichten. Die Elemente der Eingabeschicht repräsentieren den Wortschatz, über den das ACEPS verfügt. Dieser Sprachschatz kann recht große Ausmaße annehmen. Um eine Abschätzung der Größe des Wortschatzes des ACEPS vornehmen zu können, wurden einige kleinere Tests mit 85 Trainingsmails und 107 Testmails, bei denen sich schon eine passable Performanz von 42% ergab, durchgeführt. Dabei enthielt das Netz bereits nach dieser kleinen Anzahl von Trainingsbeispielen 5.500 Eingabeelemente. Deshalb wurden 1.000 willkürliche Mails der Mailing-Liste „connectionists@cs.cmu.edu“ benutzt, um die Anzahl neu hinzukommender Wörter und die Größe des sich daraus ergebenden Lexikons abzuschätzen. Nach 200 Dateien enthielt das Netzwerk bereits 10.000 verschiedene Wörter. Pro Mail waren also 50 neue Wörter hinzugekommen. Nach 800 Dateien waren es bereits 20.000 Wörter. Nach der letzten der 1.000 vom ACEPS eingelesenen Emails ergaben sich 25.458 verschiedenen Wörter. Das bedeutet also einen Zuwachs von immer noch 25 Wörtern pro Mail in den letzten 200 Emails dieser Menge, man kann also kaum von einer Sättigung sprechen. Bei den hier benutzten 946 Trainingsmails war demnach eine ähnliche Anzahl von Wörtern im Lexikon zu erwarten. Diagramm 6.1 zeigt die tatsächliche Länge einer

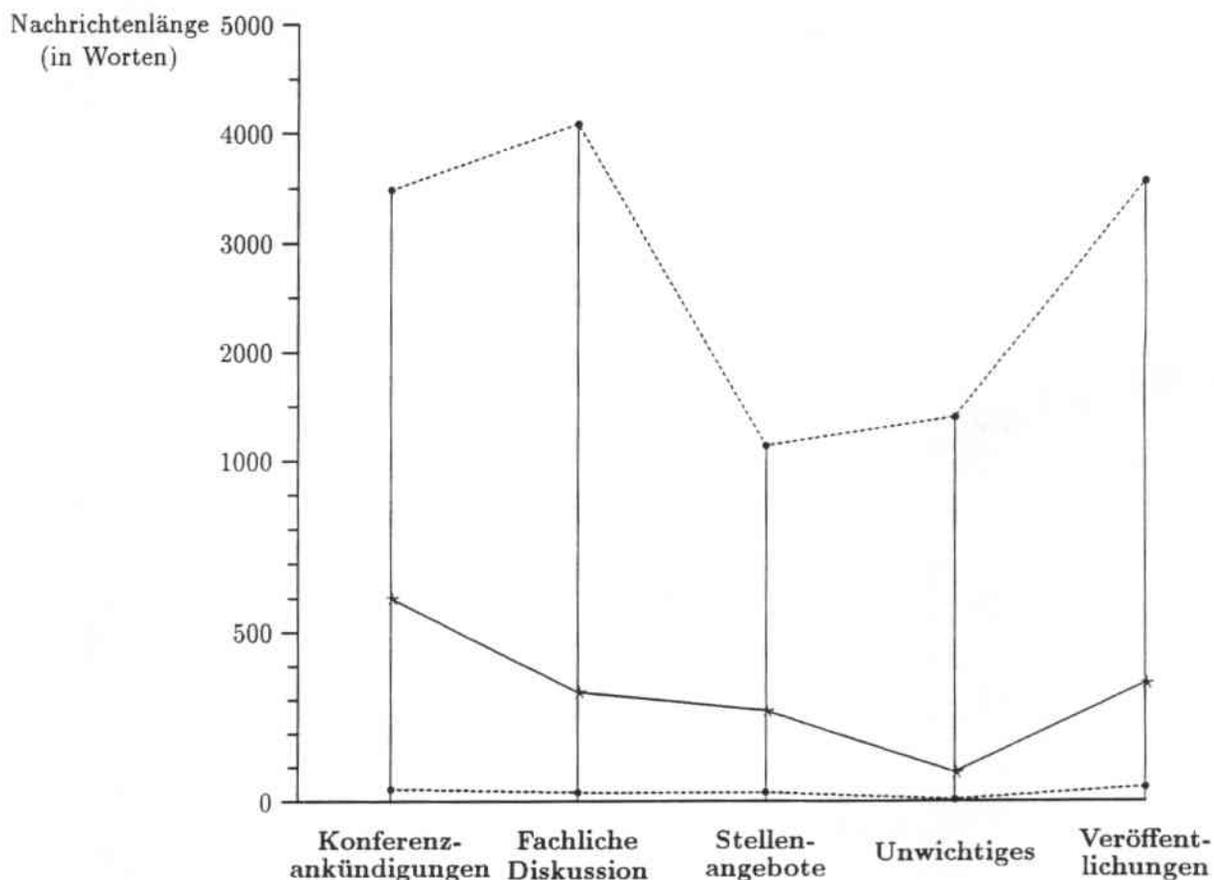


Abbildung 6.1: Länge einer Nachricht je Klasse

Nachricht wie sie im Durchschnitt in den einzelnen Klassen angetroffen wird. Dieser Wert wird graphisch durch einen Stern markiert. Außerdem sind auch minimale und maximale Wortanzahl einer Nachricht je Klasse durch Punkte markiert.

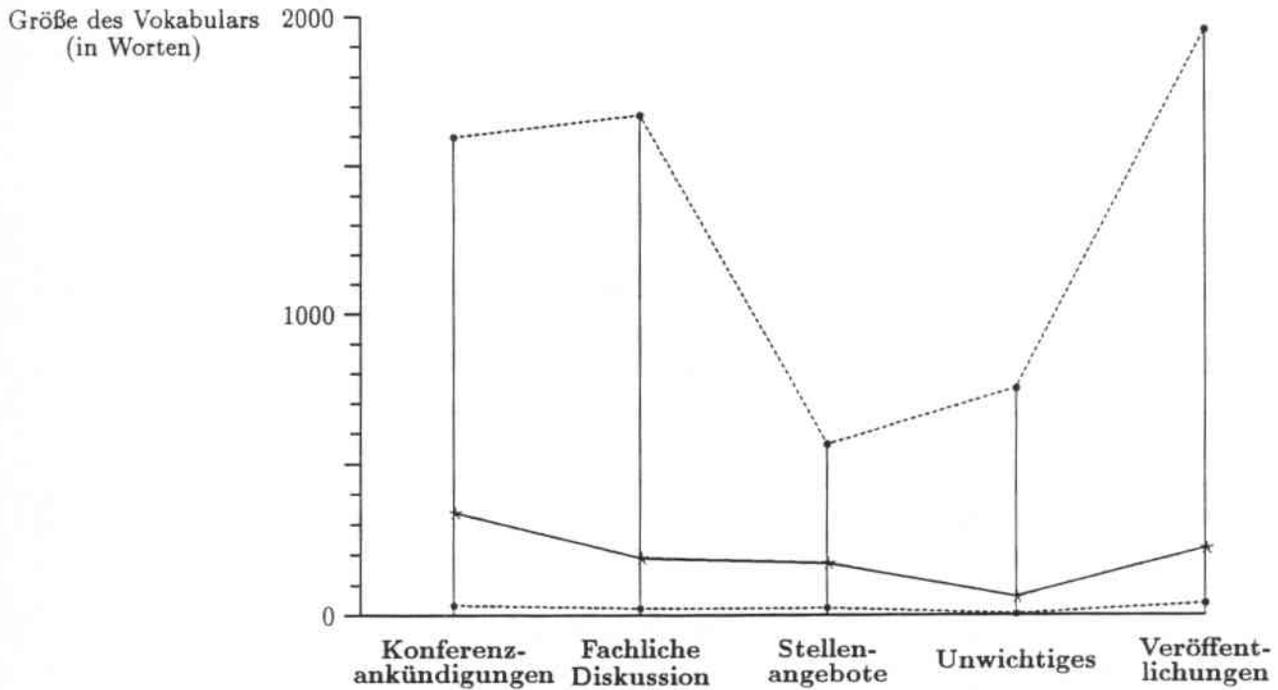


Abbildung 6.2: Größe des Vokabulars einer Nachricht je Klasse

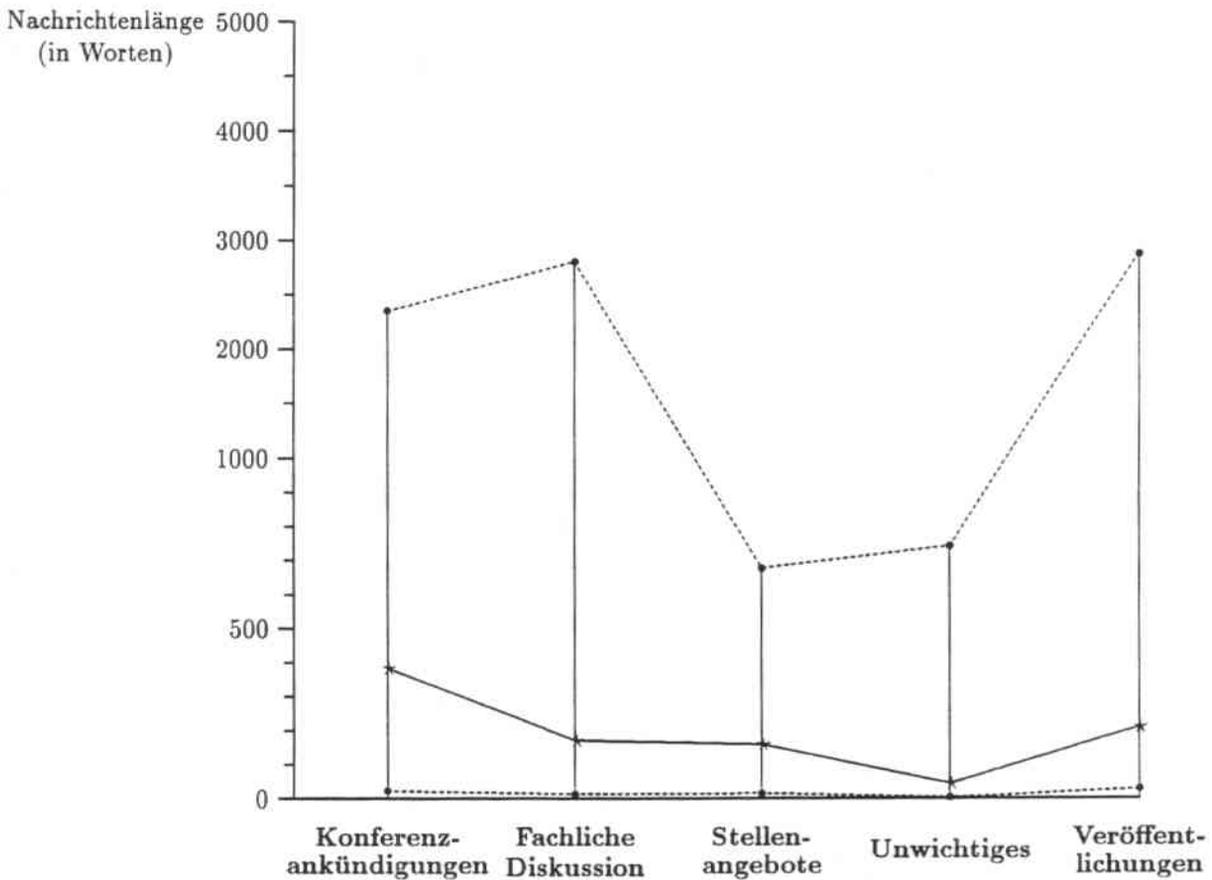


Abbildung 6.3: Länge einer Nachricht je Klasse nach Bearbeitung durch das ACEPS

Diese Darstellung gilt ebenfalls für Diagramm 6.2, das die durchschnittliche, minimale

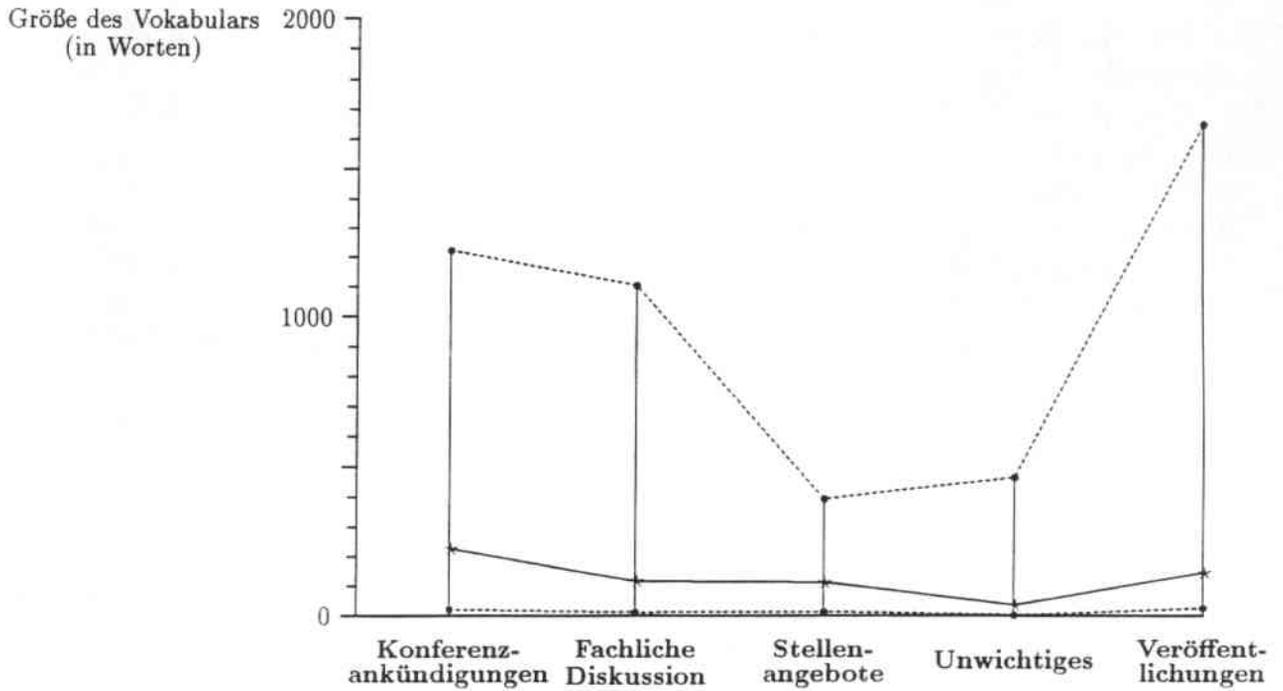


Abbildung 6.4: Größe des Vokabulars einer Nachricht je Klasse nach Bearbeitung

und maximale Größe des Vokabulars einer Email innerhalb einer Klasse repräsentiert. Die entsprechenden Diagramme werden auch für alle Emails nach Wegnahme der Leitweginformation und Herausfiltern nicht aussagekräftiger Wörter aufgestellt. Diagramm 6.3 stellt die Anzahl der Wörter pro Nachricht, Diagramm 6.4 die Größe des Vokabulars dar. Im Durchschnitt reduziert sich die Anzahl der Wörter pro Email um 42%, die Größe des Vokabulars sinkt auf durchschnittlich 64% des ursprünglichen Wertes.

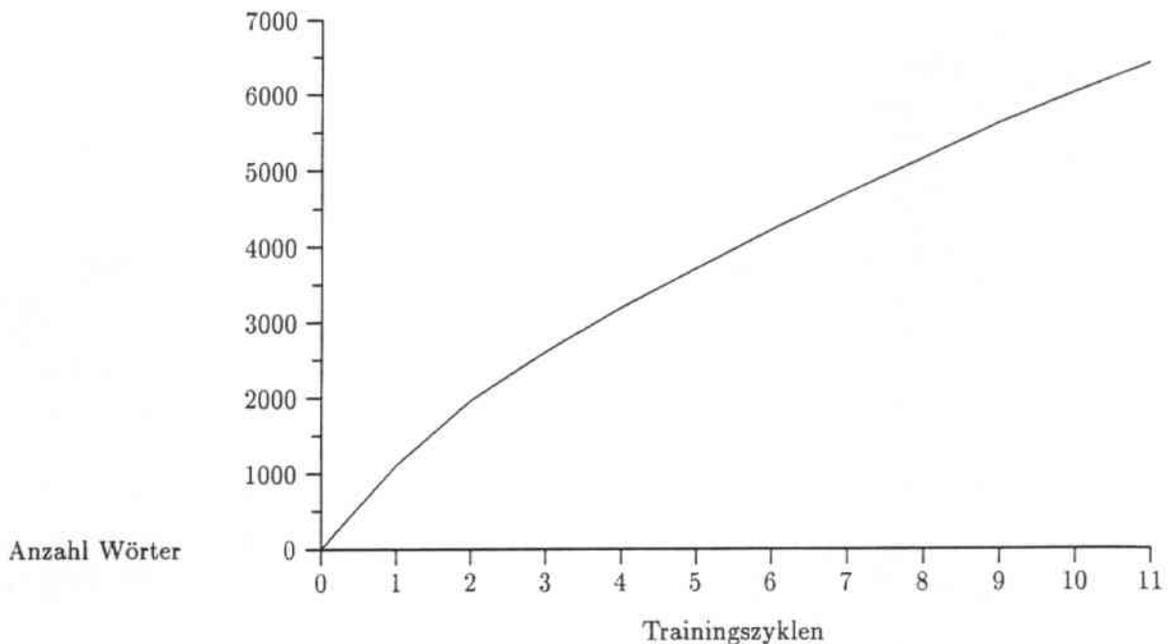


Abbildung 6.5: Wachstum des Wortschatzes des ACEPS

Bei Betrachtung des Sprachschatzes des ACEPS ist das Wachstum des Wortschatzes innerhalb der 11 Zyklen sowie die Häufigkeit des Vorkommens einzelner Wörter von besonderem Interesse. Diagramm 6.5 stellt das Wachstum des Wortschatzes vom 1. zum 11. Trainingszyklus graphisch dar. Dabei wird eine Beschränkung der 946 Trainingsmails auf die ersten 50 Worte vorausgesetzt. Es zeigt sich deutlich, daß auch im letzten Trainingszyklus noch eine erhebliche Vergrößerung des Wortschatzes stattfindet. Bei 86 neuen Mails werden durchschnittlich 4,5 neue Wörter ins ACEPS-Lexikon aufgenommen, was einer Zunahme von 390 Wörtern entspricht. Das Schaubild demonstriert, daß auch bei 946 Trainingsmails noch keine Sättigung eingetreten ist und der Wortschatz des Systems bei Hinzukommen neuer Mails weiter wachsen wird.

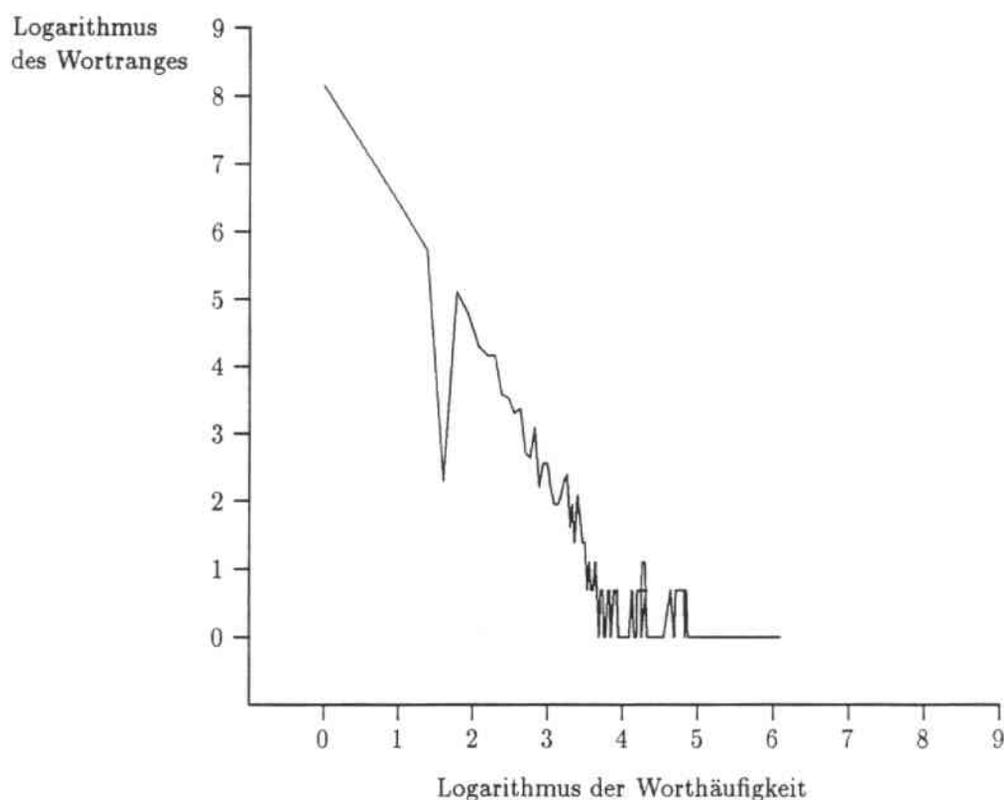


Abbildung 6.6: Rang-Häufigkeits-Verteilung von Wörtern

Bild 6.6 stellt eine Rang-Häufigkeits-Verteilung der Wörter des ACEPS-Wortschatzes dar. Die Häufigkeit eines Wortes ist als die Anzahl seines Auftretens innerhalb der 946 Trainingsmails definiert. Der Rang entspricht der Anzahl der unterschiedlichen Wörter, die diese Häufigkeit aufweisen. Es läßt sich leicht feststellen, daß einige wenige Wörter sehr oft vorkommen, während viele andere sehr selten auftauchen. Das lineare Verhalten dieser Beziehung auf einer log-log-Skala ist im Bild veranschaulicht und steht in Beziehung mit einem allgemeinen Phänomen das als *Zipfs Gesetz* [Pie61] bezeichnet wird.

Bereits in Kapitel 4 wurde erwähnt, daß im Deutschen rund 4.000 Wörter 75-80% eines normalen unspeziellen Textes ausmachen. Eine ähnliche hier durchgeführte Untersuchung berücksichtigt nicht den gesamten Wortschatz, sondern lediglich die durchschnittliche Anzahl verschiedener Wörter, die einen bestimmten Anteil eines Textes ausmachen. Dabei ergibt sich, daß in den einzelnen Themengebieten zwischen 27 (in Nachrichten der Ka-

tegorie „Unwichtiges“) und 89 (in Nachrichten der Klasse „Konferenzankündigungen“) Wörter notwendig sind, um 60% des Textes einer hier verwendeten Email darzustellen.

Nach [Moo91] besteht eine Beziehung zwischen der Generalisierungsfähigkeit eines Netzwerkes, der Menge vorhandener Trainingsmuster und der Anzahl der Parameter des Netzes. Ebenso zeigt die Erfahrung mit anderen Anwendungen, daß sich in der Regel keine gute Generalisierungsfähigkeit erreichen läßt, wenn ein Netzwerk mit ca. 26.000 Eingabe- und 5 Ausgabeelementen, d.h. 130.000 Verbindungen, durch nur 946 Muster trainiert wird. Daher muß die Zahl der Parameter so weit wie möglich eingeschränkt werden.

Eine Möglichkeit, den Wortschatz und damit auch die Größe des Netzwerkes zu begrenzen, ist, jede zu verarbeitende Email auf für den Inhalt der Nachricht relevante Wörter zu beschränken. Zum Erreichen dieses Ziels sind unterschiedliche Vorgehensweisen bzw. Kombinationen dieser Aktionen denkbar:

- Herausnahme der Leitweginformation aus einer Nachricht
- Herausfiltern der nicht aussagekräftigen Wörter aus dem Text einer Nachricht
- Herausfiltern häufig vorkommender Wörter aus dem Text einer Nachricht
- Reduzierung der Wörter einer Nachricht auf Morpheme
- Reduzierung der Wörter einer Nachricht auf lexikalisch existente Wörter

Die erste Maßnahme ist die Herausnahme der Leitweginformation, die durch das bereits erwähnte Kürzen der Nachricht um den Kopf mit Hilfe der Vorverarbeitungskomponente erfolgt. Der Absender einer Nachricht sowie die ihren Inhalt betreffenden Stichworte der „subject“-Zeile² werden beibehalten, da sonst eventuell wichtige Informationen den Sinn und Inhalt einer Nachricht betreffend verlorengehen würden. Orientiert am Verhalten eines Menschen, der beide Informationen beim Lesen einer Email berücksichtigt, bezieht auch das ACEPS nicht nur den Inhalt einer Nachricht, sondern auch Absender und Gegenstand des Textes mit in die Klassifikation ein. Strukturelle Eigenschaften eines Textes, wie z.B. die Länge einer Nachricht spielen bei der Beurteilung auch eine Rolle, ebenso z.B. eine Nachricht mit leerem Textkorpus. Dies wird bei unseren Tests jedoch nicht berücksichtigt. Die Mails haben eine durchschnittliche Länge von 326 Wörtern. Durch das Entfernen des Kopfes können bei kürzeren Emails rund 25% der Wörter eingespart werden.

Ein weiterer Schritt ist das Herausfiltern nicht aussagekräftiger Wörter durch die Liste nicht deskriptiver Wörter (LNW). Dazu gehören Wörter, die in der Mehrzahl englischer Texte anzutreffen sind und (siehe Kapitel 4.1) den Großteil eines Textes ausmachen. *A, an, the, and, for* und *with* sind nur einige Beispiel hierfür. Die gesamte Liste, die aus 341 englischen Wörtern, den Zahlen von 1 bis 100, 34 Sonderzeichen und den 26 Kleinbuchstaben *a* bis *z* besteht, befindet sich in Anhang B. Abhängig von Inhalt und Länge einer Mail sorgt diese Liste nicht deskriptiver Wörter für eine weitere Reduktion der Anzahl der Wörter um 10-20%, je nach Länge der Mail.

²Stichworte, die den Gegenstand der Nachricht zusammenfassen.

Ebenfalls eine Idee zur Reduzierung des Wortschatzes des ACEPS und damit der Parameter des Netzes ist die Herausnahme aller Wörter, die sehr häufig in Emails vorkommen. Dabei handelt es sich um Wörter, die in so großer Anzahl in allen Emails des Fachbereichs Konnektionismus auftauchen, daß sie keine Unterscheidung der hier definierten Klassen mehr zulassen. „Häufig“ wird hierbei als das Auftreten eines Wortes in mehr als 75% aller Emails definiert. Daraufhin wurden alle Emails der Trainingsmenge nach solchen Wörtern durchsucht, um eine Liste häufiger Wörter ähnlich der LNW aufzustellen. Es ergeben sich aber lediglich fünf Wörter, nämlich *neural*, *network*, *networks*, *learning* und *university*, die über 709mal in allen 946 Trainingsmustern vorhanden sind. Aufgrund dieser geringen Menge wurde die Idee einer Liste häufiger Wörter nicht verwirklicht.

Nach wie vor ist die Umgebung, in der ein Wort angetroffen wird, ein Problem und muß differenzierter betrachtet werden. Beim Einlesen eines Wortes werden alle direkt aufeinanderfolgenden Zeichen als Einheit eingelesen. Kommt ein Wort, z.B. *paper*, in seiner Originalform und in Klammern (*paper*) oder Hochkommata '*paper*' in einem Text vor, handelt es sich dabei eigentlich auch um zwei verschiedene Wörter. Um nun aber zu verhindern, daß zwei verschiedene Elemente im Netzwerk geschaffen werden, müssen Zeichen wie Klammern und Hochkommata am Anfang oder Ende eines Wortes automatisch entfernt werden. Dies kann als eine Art Rückführung von Wörtern in ihre Originalform betrachtet werden, läßt aber immer noch verschiedene grammatikalische Formen eines Wortes zu. So werden im Netz immer noch zwei Elemente für Singular und Plural eines Wortes geschaffen: *paper* und *papers*. Mit einem automatischen Abstreichen des Wortendes *-s* ist es dabei nicht getan, gibt es doch auch Wörter, die tatsächlich mit einem *-s* enden und keinesfalls der Pluralform eines Substantives entsprechen wie z.B. *delicious*. Das gleiche gilt für die verschiedenen Konjugationen *get*, *gets*, *getting* und *got* des Verbes *to get*. Auch hier läßt sich nicht durch bloßes Abstreichen einer Endung die Grundform herstellen. Dazu benötigt man ein Lexikon und grammatikalische Regeln, die mit Hilfe eines Parsers (siehe Abschnitt 4.3) die jeweilige Grundform eines Wortes bestimmen. Durch Reduzierung aller Wörter auf ihre Morpheme läßt sich ebenfalls eine Verringerung der Größe des Wortschatzes erreichen. In Abschnitt 6.2.8 findet man eine genaue Beschreibung sowie die Testergebnisse eines Experiments, in dem mit Hilfe des LA-MORPH Systems nur Grundformen von Wörtern als Elemente in die Eingabeschicht des Netzwerkes aufgenommen wurden.

Ein weiteres Problem sind auch Sonderzeichen, die innerhalb eines Wortes vorkommen oder Tippfehler, bei denen Buchstaben vertauscht, ausgelassen oder hinzugefügt werden. Diese Worte wurden nicht herausgefiltert, sondern in den meisten Simulationen wie sinnvolle Wörter behandelt. Lediglich in einem Experiment (siehe Abschnitt 6.2.7) wurde der Wortschatz des ACEPS auf lexikalisch existente Wörter des englischen Sprachschatzes beschränkt.

6.2 Verwendete Verfahren

Im folgenden Abschnitt werden alle im Rahmen dieser Arbeit mit dem ACEPS ausgeführten Simulationen³ beschrieben. Das mit fünf Testpersonen durchgeführte Experiment (MENSCH) aus Abschnitt 6.1.3 dient dabei als Vergleichswert für die Leistungsfähigkeit des ACEPS.

In Abschnitt 6.1.2 wurde der Ausdruck „Zyklustest“ definiert. Die Ergebnisse dieses Zyklustests werden im folgenden nicht erwähnt werden, da lediglich eine verwirrend große Anzahl von Simulationsergebnissen entstand, die relativ zueinander nicht von den regulären Test- und Trainingsergebnissen abweichen. Allgemein konnten durch die Zyklustests aber bis zu 10% bessere Ergebnisse als mit der offiziellen Testmenge erreicht werden.

6.2.1 Einfache Klassifikation durch das ACEPS

Im ersten Versuch, der mit dem ACEPS durchgeführt wurde, werden die 946 Emails der Trainingsmenge in ihrer vollen Länge als Trainingsdaten verwendet und das Netz mit Hilfe des adaptiven Trainingsalgorithmus trainiert (ADPT). Wie in allen folgenden Simulationen⁴ werden der aus dem Originalkopf der Email übriggebliebene Gegenstand und Absender wie alle anderen Worte des Textkorpus der Email behandelt. Das ein- oder mehrmalige Vorkommen eines Wortes innerhalb einer Nachricht sorgt für die Aktivierung des entsprechenden Eingabeelements des ACEPS-Netzwerkes. Dabei ergibt sich in 11 Trainingszyklen auf der Menge der Testmails eine durchschnittliche Performanz von 46,6%, angefangen bei 46,1% nach dem ersten Training über einen Höchststand von 48,1% nach der 7. Trainingseinheit bis hin zu 47,3% im letzten Trainingszyklus. Die Größe des Lexikons wächst bei 946 Eingabenachrichten auf 26.470 Worte, aus denen dieselbe Anzahl von Eingabeelementen im Netzwerk resultiert. Daraus ergibt sich bei 5 Ausgabeelementen des Netzwerkes eine Menge von 132.350 Verbindungen.

Anstelle der einmaligen Aktivierung eines Eingabeelements gibt es auch die Möglichkeit, die Häufigkeit des Vorkommens eines Wortes zu zählen und die Aktivierung des Eingabeelements (= 1), das dieses Wort repräsentiert, mit der sich ergebenden Zahl zu multiplizieren. Dieser Versuch, der die Worthäufigkeit berücksichtigt (ZÄHL), wurde als zweite Simulation durchgeführt, bringt aber keine besseren Ergebnisse. Die Performanz des Netzes auf der Testmenge beginnt bei 44,2% und steigt bis zu 46,9% im letzten Zyklus.

Die gelernten Trainingsmuster erweisen sich in beiden Fällen als nicht stabil. Die 11 Zyklen ergeben eine durchschnittliche Genauigkeit von 85,7% bzw. 84,9%, mit der die Menge der Trainingsdaten richtig klassifiziert wird. Nach dem ersten Training mit 86 Emails kann sich das Netz im ersten Test noch an 83 der gelernten Muster erinnern, was einer Genauigkeitsrate von 96,5% entspricht. Bei Berücksichtigung der Häufigkeit des Vorkommens von Wörtern innerhalb eines Tests auf diesen 86 Trainingsdaten wird sogar noch ein Muster weniger richtig klassifiziert, was zu einer Rate von 95,3% führt.

³Einige wenige Tests finden keine Erwähnung, da sie sich nur unwesentlich von anderen unterscheiden und keine Leistungsverbesserungen bewirken.

⁴Ausgenommen die Realisierung von 3 Teilnetzen in Abschnitt 6.2.3.

Nach 11 Zyklen, in denen 946 Muster gelernt werden, sinkt die Leistung auf den Trainingsdaten auf 76,7% bzw. 76,1%. Die Resultate der beiden Testfälle (ADPT und ZÄHL) sind noch einmal in Tabelle 6.14 zusammengefaßt.

Beschreibung	Anzahl Wörter	Anzahl Verbindungen	Trainingsleistung	Testleistung
ADPT - Einfache Klassifikation	26.470	132.350	76,7%	47,3%
ZÄHL - Einfache Klassifikation - Worthäufigkeit	26.470	132.350	76,1%	46,9%

Tabelle 6.14: Ergebnisse einer einfachen Klassifikation durch das ACEPS

Die Tabelle ist folgendermaßen aufgebaut: zunächst wird die Art des ausgeführten Experiments beschrieben. In der zweiten Spalte findet sich die daraus resultierende Anzahl der Wörter im Lexikon des ACEPS, die auch der Zahl der Eingabeelemente des Netzwerkes entspricht. Nach der Anzahl der Verbindungen zwischen allen Einheiten des Netzes erscheint die Genauigkeitsrate des Netzwerkes auf der Menge der Trainingsmuster gefolgt von einer Klassifikation aller Testdaten. Der Aufbau aller folgenden Tabellen in den nächsten Abschnitten dieses Kapitels entspricht im wesentlichen dieser Struktur, weshalb sie im folgenden nicht noch einmal erläutert wird.

Wie schon erwähnt findet in den 11 Zyklen ein kontinuierlicher Leistungsabfall auf den Trainingsmustern statt, je mehr Wissen sich das Netz erwirbt. Ebenso zeigt das Netz mangelhafte Generalisierungsfähigkeit auf den Testdaten. Oberste Priorität ist folglich nicht, die Performanz auf der Menge der Trainingsmuster zu steigern, sondern die Leistung des ACEPS auf neuen noch nie gesehenen Testmails zu verbessern. Mit einer Verringerung der Anzahl der Netzparameter ist implizit auch ein besseres Lernen des Netzes verbunden. Da jedes Eingabeelement des ACEPS-Netzwerkes ein Wort des ACEPS-Lexikons repräsentiert, wird die Reduktion der Parameter durch eine Einschränkung der Größe des Wortschatzes des ACEPS erreicht. Die nächsten beiden Abschnitte suchen nach Wegen, weitere nicht relevante Wörter aus den Texten herauszufiltern und so die Größe des Netzes des ACEPS zu reduzieren.

6.2.2 Beschränkung der Eingabe

Die zweifellos einfachste Methode, den Wortschatz des ACEPS zu verringern und sein Netzwerk möglichst klein zu halten, ist eine radikale Kürzung des Eingabetextes auf eine bestimmte Anzahl von Wörtern. Dazu wurden verschiedene Experimente mit unterschiedlichen Eingabelängen der Emails durchgeführt. Tabelle 6.15 präsentiert die einzelnen Simulationsergebnisse und die sich ergebende Anzahl der Wörter im Lexikon bei einer Beschränkung der Emails auf Längen von 10, 30, 50, 70 und 90 Worte (ADPT_ANF10, ADPT_ANF30, ADPT_ANF50, ADPT_ANF70, ADPT_ANF90).

Beschreibung	Anzahl Wörter	Reduktion des Wortschatzes	Anzahl Verbindungen	Trainingsleistung	Testleistung
ADPT - Einfache Klassifikation	26.470	—	132.350	76,7%	47,3%
ADPT_ANF10 - Beschränkung auf 10 Worte	1.930	92,7%	9.650	93,0%	62,8%
ADPT_ANF30 - Beschränkung auf 30 Worte	4.475	83,1%	22.375	96,5%	69,0%
ADPT_ANF50 - Beschränkung auf 50 Worte	6.428	75,7%	32.140	96,0%	72,1%
ADPT_ANF70 - Beschränkung auf 70 Worte	7.939	70,0%	39.695	96,7%	70,2%
ADPT_ANF90 - Beschränkung auf 90 Worte	9.137	65,5%	45.685	96,4%	67,8%
ZÄHL_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit	6.428	75,7%	32.140	96,6%	72,9%
ADPT_WORT5 - Einschränkung auf 5 Buchstaben	17.541	33,7%	87.705	63,0%	41,9%

Tabelle 6.15: Simulationsergebnisse bei Beschränkung des Eingabetextes

Schon nach dem ersten Trainingszyklus ergeben sich in allen fünf Fällen bessere Leistungen als in den zuvor realisierten Simulationen. Ebenso ist zwar noch immer ein Abfall der Trainingsleistung innerhalb der 11 Zyklen zu verzeichnen, jedoch beträgt der Unterschied von der ersten zur letzten Lernphase 2,2 - 4,7% , nicht mehr beinahe 20% wie in den in Abschnitt 6.2.1 durchgeführten Simulationen ADPT und ZÄHL. Die dort aufgestellte These einer zu großen Anzahl von Verbindungen für die Menge zur Verfügung stehender Trainingsmuster bewahrheitet sich hier.

Durch die vorgenommene Einschränkung der Eingabe auf Mailanfänge verkleinert sich der Wortschatz des Systems und damit das Netzwerk um 65% bis beinahe 93%! Entsprechend nimmt auch die Anzahl der Verbindungen im Netzwerk ab. Diese Reduzierung führt zu Genauigkeitsraten der verschiedenen Ausprägungen des ACEPS, die zwischen 62,8% und 72,1% liegen. Diagramm 6.7 stellt die Leistung der fünf Systeme graphisch dar.

Die durchgehende Linie gibt die Ergebnisse der Testmenge, die gestrichelte die der Trainingsmenge an. Aus den durch die fünf Experimente entstandenen Netzwerken wird nun das mit der höchsten Performanz ausgesucht. Bei einer Eingabelänge von 50 Worten wird

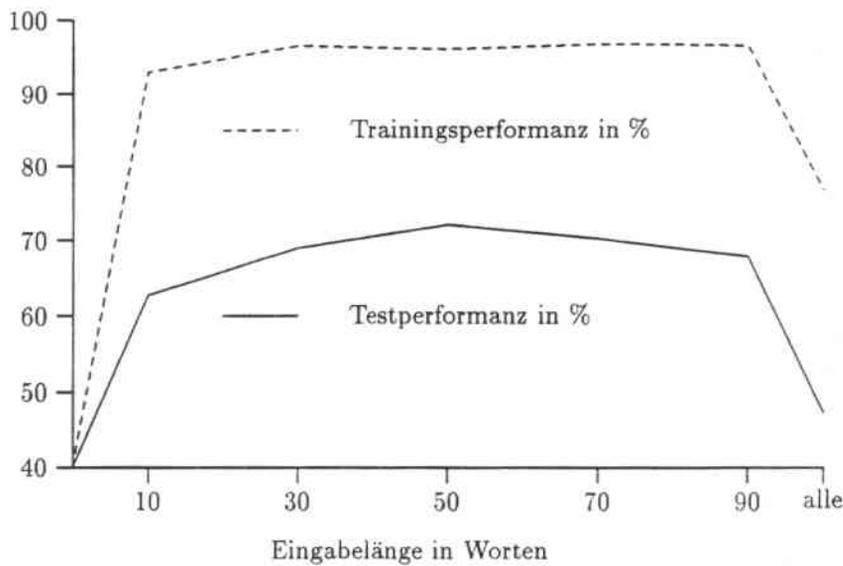


Abbildung 6.7: Vergleich der Performanz verschiedener Eingabelängen

noch einmal die Häufigkeit eines Wortes innerhalb einer Email berücksichtigt. Bei diesem Experiment (ZÄHLANF50) ergeben sich bessere Ergebnisse als beim vergleichbaren Versuch mit Emails der vollen Länge.

Tabelle 6.15 enthält auch die Ergebnisse dieser Simulation. Bei einer Größe des Lexikons von 6.428 Wörtern wird hier die bisher beste Leistung mit 72,9% Genauigkeit auf der Testmenge erzielt und auch die Performanz von 96,6% auf den Trainingsdaten entspricht beinahe der besten Leistung, die bisher nach 11 Zyklen erbracht wurde. Die Reduzierung der Anzahl von Verbindungen hat dafür gesorgt, daß die in den Emails vorkommenden Wörter stärkere Verbindungen zu den „richtigen“ Klassen aufbauen. Bei Berücksichtigung der gesamten Nachrichten werden viele Wörter in allen Klassen beinahe gleichverteilt angetroffen, so daß sie nicht mehr zu einer richtigen Einordnung beitragen.

Eine weitere denkbare Beschränkung der Eingabe des ACEPS ist das Kürzen eines Wortes auf den Wortanfang. Ziel dieser Handlungsweise ist letztendlich ebenfalls eine Verringerung des Wortschatzes des ACEPS. Läßt man nur eine bestimmte Anzahl von Buchstaben in jedem Wort des Lexikons zu, besteht das Sprachmodell nur aus Wortanfängen. Diese Wortanfänge müssen über eine Länge verfügen, die einerseits immer noch eine gewisse Semantik beinhaltet, andererseits dürfen sie nicht zu lang sein, um eine tatsächliche Reduzierung des Wortschatzes zu gewährleisten. Da die Wörter des englischen Wortschatzes in der Regel erheblich kürzer sind als deutsche Wörter, wird im nachfolgenden Experiment die zulässige Länge eines Wortes auf 5 Zeichen bzw. Buchstaben begrenzt. Die Auswahl gerade dieses Wertes läßt sich anhand eines einfachen Beispiels erklären: bei vielen englischen Verben besteht die Grundform des Wortes aus 5 oder weniger Buchstaben. Dazu gehören *learn*, *sleep*, *take* usw. Berücksichtigt man nur die ersten 5 Buchstaben dieser Wörter wird z.B. im Falle von *learn* für die fünf Wörter *learn*, *learns*, *learning*, *learned* und *learner* nur ein Netzelement anstelle von fünf derselben Bedeutung erzeugt. Wird ein Wert größer 5 gewählt, tritt keine entscheidende Ersparnis von Elementen und Verbindungen mehr ein. Ein kleinerer Wert zerstört die Semantik sehr vieler Wortanfänge, da zu viele

verschiedene Wörter unterschiedlicher Bedeutung auf dasselbe Netzelement abgebildet werden. Trotzdem mag auch schon bei dem Versuch mit Wortlänge 5 gerade bei Substantiven der englischen Sprache ein gemeinsamer Wortanfang von Wörtern entstanden sein, deren Semantik keinerlei Verwandtschaft aufweist. Da auch die Simulationsergebnisse keine Verbesserung der bisher optimalsten Ausprägung des ACEPS ergeben, wurde dieses Experiment (ADPT_WORT5) auch nicht mit einer anderen Anzahl von Zeichen weiter fortgesetzt. Die Resultate des Versuchs finden sich in Tabelle 6.15.

6.2.3 Realisierung von 3 Teilnetzen

Weniger zur Verringerung der Parameter innerhalb des Netzwerkes, sondern eher zur Steigerung der Leistungsfähigkeit des ACEPS werden die in diesem Abschnitt beschriebenen Experimente durchgeführt. Dabei wird anstelle eines einzigen Netzwerkes, in dem Gegenstand, Absender und Textkorporus einer Email repräsentiert werden, ein Netzwerk geschaffen, das aus 3 voneinander abgegrenzten Teilnetzen besteht. In jedem Teilnetz wird der Inhalt eines bestimmten Abschnitts aller bisher gelernten Nachrichten dargestellt:

1. Die Eingabeschicht des ersten Netzwerkes enthält Elemente für alle Wörter, die jemals in der „subject“-Zeile einer der 946 Trainingsmails als Thema der Nachricht angegeben wurden.
2. Das zweite Teilnetz besteht in der Eingabeschicht aus Elementen, die für die Absender der 946 Mails stehen.
3. Im dritten Teilnetz schließlich werden durch die Eingabeknoten die Wörter des Textkorporus der Mail repräsentiert.

Jedes der drei Teilnetze verfügt über fünf „Ausgabe“-Elemente, die für die fünf Klassen der hier verwendeten benutzerspezifischen Variante des ACEPS stehen. Die Aktivierungen der Elemente dieser Zwischenschicht des ACEPS-Netzwerkes werden benutzt, um die Aktivierung der tatsächlichen Ausgabeelemente des Netzwerkes zu berechnen.

Im ersten Fall (BIGNET) geschieht dies durch einfache Aufsummation der Aktivierung der „Ausgabe“-Elemente jedes Teilnetzes. Diese gleichberechtigte Miteinbeziehung der Aktivierung aller 3 Teilnetze ist sicherlich keine optimale Kombination. Um die aufwendige Ermittlung einer solchen bestmöglichen Verbindung mittels komplizierter und zeitaufwendiger Berechnungen zu vermeiden, wurde eine Lösung gewählt, die sich folgendermaßen begründen läßt: Zur Klassifikation einer Email dieses Themenbereichs ist es von geringer Bedeutung, wer der Absender einer Email ist⁵. Eine flüchtige Durchsicht der „subject“-Zeilen der vorliegenden Trainingsmails zeigte, daß das Thema einer Nachricht dort (leider) oft auch nur recht unvollkommen zusammengefaßt ist. In den meisten Fällen läßt sich aus den in der „subject“-Zeile der Nachricht vorkommenden Schwörtern nicht

⁵Bei einer anderen Unterteilung der Klassen oder Prioritätsstufen des ACEPS kann der Absender eine erheblich größere Rolle spielen. Dann nämlich, wenn zwischen Geschäfts- und privaten Nachrichten unterschieden wird, und schon die Adresse des Absenders mit hoher Wahrscheinlichkeit zur Einordnung in eine bestimmte Klasse führt.

unbedingt auf den Inhalt schließen. Der eigentliche Textkorpus bleibt zur Einordnung der Nachricht also immer das wichtigste Kriterium. In unserem Versuch (BIGNET_LK) ergibt sich die Aktivierung der Ausgabeelemente des Gesamtnetzes aus einer Linearkombination der 3 Teilnetze, bei der der Textkorpus am höchsten gewichtet wird, gefolgt vom Gegenstand der Nachricht. Die Aktivierung des dritten Teilnetzes mit den Absendern geht mit geringem Anteil in die Gesamtaktivierung des Netzes ein.

In beiden Experimenten wird der Textkorpus der Email in voller Länge berücksichtigt. Bei gleichberechtigter Berücksichtigung der Ergebnisse aller 3 Teilnetze (BIGNET) erreicht das Netz des ACEPS eine Leistung von 49,6% auf der Menge der Testmails. Das zweite Experiment (BIGNET_LK) erzielt ein Ergebnis von 48,8%.

Gegenüber dem ersten Versuch (ADPT) in Abschnitt 6.2.1, bei dem nur ein Netzwerk realisiert wurde, ist dies eine Verbesserung von 2,3% bzw. 1,5%. Ebenso erhöht sich die Performanz auf den Trainingsdaten in beiden Fällen um 1,9% auf 78,6% nach 11 Zyklen. Zwar klassifizieren die in diesem Versuch entstandenen Netzwerke die Trainingsmuster schon im 1. Zyklus mit höherer Genauigkeit als im entsprechenden vergleichbaren Versuch — ein detaillierter Vergleich aller Ergebnisse findet sich in Tabelle 6.16 — jedoch nimmt die Trainingsleistung innerhalb 11 Zyklen noch immer um fast 20% ab.

Beschreibung	Anzahl Wörter	Anzahl Verbindungen	Trainingsleistung	Testleistung
ADPT - Einfache Klassifikation	26.470	132.350	76,7%	47,3%
BIGNET - 3 Teilnetze	26.470	132.350	78,6%	49,6%
BIGNET_LK - 3 Teilnetze - Linearkombination	26.470	132.350	78,6%	48,8%
BIGNET_ANF50 - Beschränkung auf 50 Worte - 3 Teilnetze	6.428	32.140	97,6%	67,4%

Tabelle 6.16: Testergebnisse einer einfachen Klassifikation mit 3 Teilnetzen

Im vorigen Abschnitt ließ sich das Problem auf 2-5% reduzieren. Dies läßt sich damit erklären, daß die Eingabe des Netzwerkes radikal eingeschränkt wurde. Zum Vergleich wird auch ein Versuch mit 3 Teilnetzen durchgeführt, der Mails nur bis zu einer Länge von 50 Wörtern berücksichtigt (BIGNET_ANF50). Das Ergebnis ist dabei 67,4% auf den Test- und 97,6% auf den Trainingsdaten. Dabei zeigt sich deutlich, daß die Trainingsleistung des Netzes rapide abnimmt, je größer der Wortschatz und damit die Zahl der Parameter des Systems wird. Nach dem ersten und zweiten Trainingszyklus erinnert sich das Netzwerk nämlich noch an alle 86 bzw. 172 gelernten Trainingsmuster. Erst nach dem 3. Trainingslauf fällt die Genauigkeit von 100% auf 99,2%. Sie verschlechtert sich in allen übrigen Zyklen und resultiert schließlich in den bereits erwähnten 97,6%.

In einer praktischen Anwendung des ACEPS ist ein solches Verhalten nicht tolerabel. Zwar mag ein Benutzer des Systems nicht sofort bei seiner ersten Anwendung über eine

große Menge vorklassifizierter Trainingsmuster verfügen, jedoch sammelt sich im Laufe der Zeit eine sichtlich größere Anzahl als nur 946 Emails an⁶. Diese Zahl vergrößert das Lexikon des ACEPS und die *Stabilität* des Netzwerkes nimmt eventuell noch weiter ab. Stabilität bedeutet in diesem Zusammenhang, daß einmal gelernte Muster vom Netzwerk erinnert werden sollen, auch wenn immer wieder neue Muster gelernt werden. In vielen Fällen destabilisieren neu gelernte Muster die früher einmal gelernten. Genau dies scheint auch hier der Fall zu sein. Im nächsten Abschnitt wird mit einer Vorgehensweise experimentiert, die sich weniger mit einer allgemeinen Leistungssteigerung des Netzes befaßt. Vielmehr wird versucht, die Stabilität des ACEPS zu erhöhen und dafür zu sorgen, daß einmal erworbenes Wissen nicht wieder verlorengeht⁷.

6.2.4 Hinzunahme von Wortpaaren

Die Erweiterung des ACEPS-Netzwerkes durch Einfügen einer Zwischenschicht ist schon in Abschnitt 5.1 besprochen worden, auch die Tatsache, daß deren Elemente Wortpaare repräsentieren. Bei Berücksichtigung der vollen Länge einer Email und aller möglichen Wortpaare ergibt sich bei einer Größe des Lexikons von 26.470 Wörtern eine Anzahl von 700.660.900 neu hinzukommenden Elementen in der Zwischenschicht. Multipliziert mit der Anzahl der Elemente in der Ausgabeschicht resultiert dies in über 3 Milliarden⁸ Verbindungen. Diese riesige Zahl macht die Utopie eines solchen Planes deutlich. Selbst wenn dem ACEPS nur die Anfänge der Trainingsmails als Eingabe präsentiert werden, würden sich bei 6.428 Wörtern noch über 206 Millionen neuer Verbindungen ergeben.

Als Strategie zur Reduzierung dieser Zahl wird folgende Vorgehensweise gewählt: Es werden nur Wortpaare von Worten ins Netz aufgenommen, die in einer Trainingsmail vorkommen, die auch nachdem sie gelernt wurde, noch falsch klassifiziert wird. Damit sind zwei Voraussetzungen erfüllt. Zum einen wird die Anzahl der neuen Elemente sinnvoll eingeschränkt, zum anderen läßt diese Methode eine Verbesserung der Trainingsleistung erwarten. Der Grund, daß auch in dieser Simulation nur die ersten 50 Worte jeder Mail berücksichtigt werden, liegt in der Dauer des Lernvorganges begründet: Wurden Nachrichten dem ACEPS vollständig dargeboten, beträgt die Dauer einer Simulation über 11 Trainingseinheiten in manchen Testfällen 2-3 Tage⁹. Bei Kürzung der Mails sind die Simulationen in der Regel nach spätestens 24 Stunden beendet.

Tatsächlich ergibt sich in diesem Experiment (WP_ANF50) eine Performanz von 99,5% auf der Menge der Trainingsdaten. Die Diskrepanz von 2-20% Leistungsver schlechterung von der ersten zur letzten Trainingseinheit in den bisherigen Tests wird dadurch also nicht

⁶Dies trifft insbesondere zu, berücksichtigt man die Tatsache, daß Nachrichten einer Mailing-Liste in einer „offiziellen“ Sprache — zumeist Englisch — verfaßt sind und ein Benutzer des ACEPS außerdem noch Emails in seiner Landessprache erhält. Damit ergibt sich ein Wortschatz, der aus 2 Sprachen besteht und dementsprechend groß werden kann.

⁷Auch Backpropagation-Netzwerke versuchen dieser Tatsache durch häufiges zyklisch wiederholtes Lernen zu begegnen. Die Beschreibung eines Experiments des ACEPS mit einem Backpropagation-Algorithmus als Trainingsprozedur findet sich in Abschnitt 6.2.11.

⁸Die genaue Anzahl lautet 3.503.304.500.

⁹Auf einer DEC-Station 5000/125.

verringert, sondern die Leistung des ACEPS verbessert sich sogar um 0,7% von anfänglich 98,8% auf 99,5%¹⁰.

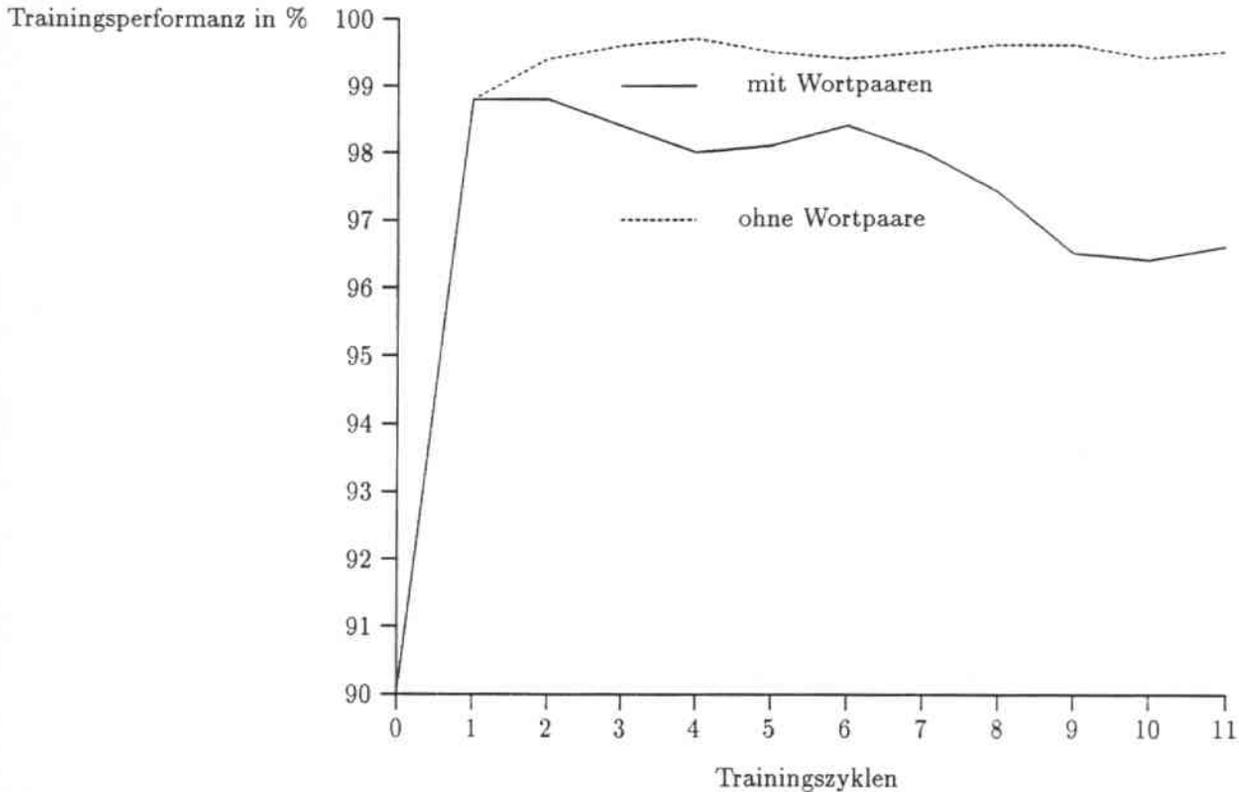


Abbildung 6.8: Vergleich der Trainingsleistung mit und ohne Hinzunahme von Wortpaaren

Durch Einfügen einer Zwischenschicht ins Netz, die korrekatives Training ermöglicht, ist also eine beinahe 100%ige Beständigkeit des gelernten Wissens des ACEPS möglich! Tabelle 6.17 veranschaulicht die genauen Simulationsergebnisse sowie die Größe des entstehenden Netzwerkes.

Eine mit dieser Erhöhung der Trainingsleistung einhergehende Verbesserung der Testleistung ist leider nicht zu verzeichnen. Angefangen bei 60,5% nach dem ersten Training verbessert sich die Leistung bis auf 72,1% nach dem letzten Trainingszyklus. Dies ist sogar etwas schlechter als das bisher erreichte Ergebnis (ZÄHL_ANF50) aus Abschnitt 6.2.2, von den 258 Testmails werden jetzt 2 mehr falsch eingeordnet als es vorher der Fall war. Da es sich dabei aber um eine so geringe Anzahl neuer Falschklassifikationen handelt, werden diese Mails nicht weiter analysiert.

Damit zeigt sich, daß die Hinzunahme von Wortpaaren zwar das Lernen einzelner Ein-/Ausgabepaare erleichtert, aber nicht zur Generalisierungsfähigkeit des Netzes beiträgt. Andere Möglichkeiten müssen gefunden werden, um dieses Ziel zu erreichen.

¹⁰Ein Vergleich der Trainingsergebnisse vor und nach der Hinzunahme von Wortpaaren ist in Bild 6.8 dargestellt.

Beschreibung	Anzahl Wörter	Anzahl Wortpaare	Anzahl Verbindungen	Trainingsleistung	Testleistung
ZÄHL_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit	6.428	—	32.140	96,5%	72,9%
WP_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare	6.428	2.166	42.970	99,5%	72,1%

Tabelle 6.17: Hinzunahme von Wortpaaren beim Training

6.2.5 Constructive Learning

Bisher wurde das Netzwerk des ACEPS-Systems folgendermaßen trainiert:

- Die Strukturen der Emails, die in einem Zyklus als kombinierte Test-/Trainingsdaten verwendet wurden, werden ins Netzwerk aufgenommen, d.h. das existierende Lexikon wird um alle Wörter einer Email erweitert, sofern sie nicht schon vorhanden sind, und für jedes Wort ein Element im Netzwerk geschaffen
- Im folgenden Trainingslauf werden dann die Gewichte dieser neuen Elemente eingestellt bzw. die Gewichte zu bereits im Lexikon vorhandenen Wörtern neu angepaßt.

Ein weiterer Ansatz, die Strukturen eines Netzwerkes aufzubauen und zu trainieren, ist der des *constructive learning*. Grundgedanke dieser Art des Lernens ist es, vorklassifizierte Ein-/Ausgabepaare nur dann in die Menge der tatsächlich vom System benutzten Trainingsmuster einzufügen, wenn das Wissen des ACEPS dadurch vergrößert werden kann. Kann das ACEPS durch ein Trainingsbeispiel nichts Neues dazulernen, d.h. verfügt es bereits über die notwendigen Kenntnisse, um eine richtige Klassifikation vorzunehmen, werden weder Lexikon noch Netz erweitert. Damit wird die Größe des Lexikons und auch die Komplexität der Netzstruktur möglichst klein gehalten. Man kann davon ausgehen, daß die in einer auf Anhieb richtig klassifizierten Email vorkommenden Wörter entweder schon alle im Lexikon, und damit auch ins Netz aufgenommen wurden oder daß zumindest die zur Klassifikation der Email relevanten Wörter bereits vorhanden sind. Es wäre daher nicht effizient, auch die restlichen, vielleicht für eine richtige Klassifikation völlig überflüssigen Wörter, ins Lexikon und somit ins Netz aufzunehmen, und damit das Volumen beider nur unnötig aufzublähen.

Beim hier durchgeführten Experiment (CLEARN_ANF50) werden wieder nur Mailanfänge in Form der ersten 50 Worte einer Nachricht berücksichtigt. Ebenso werden benachbarte Wortpaare falsch klassifizierter Trainingsmails in das Netzwerk aufgenommen. Die Vorgehensweise, nur die Strukturen falsch klassifizierter Mails ins Netzwerk aufzunehmen,

resultiert in einer Verringerung des Lexikons auf 57% seines ursprünglichen Volumens. Lexikon und Eingabeelemente des Netzes werden also auf fast die Hälfte reduziert und enthalten noch jeweils 3.669 Elemente. Tabelle 6.18 enthält die Anzahl der Wortpaare und aller Verbindungen des Netzwerkes sowie die Ergebnisse der durchgeführten Simulation.

Beschreibung	Anzahl Wörter	Anzahl Wortpaare	Anzahl Verbindungen	Trainingsleistung	Testleistung
WP_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare	6.428	2.166	42.970	99,5%	72,1%
CLEARN_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - <i>constructive learning</i>	3.669	243	19.560	100%	64,0%
WEIGHT_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - Manipulation von Verbindungen	6.428	2160	42.940	99,8%	72,9%

Tabelle 6.18: Klassifikation durch das ACEPS mit *constructive learning* und Manipulation von Verbindungen

Die Strategie des *constructive learning* ermöglicht, auch die 5 Trainingsmails zu lernen, die bisher selbst von der leistungsfähigsten Variante des ACEPS-Netzwerkes (ZÄHL_ANF50) falsch klassifiziert wurden. Die Performanz auf der Menge der Trainingsmuster steigt von 98,5% auf 100% schon nach dem 3. Trainingszyklus. Diese Leistung bleibt bis zum letzten Trainingslauf stabil.

Gleichzeitig ergibt sich eine deutlich schlechtere Leistung bei der Klassifikation der Testmenge. Besonders deutlich wird dies nach dem ersten Trainingszyklus, wo im Experiment mit *constructive learning* eine 15% schlechtere Leistung, nämlich 45,7% Genauigkeit anstelle von 60,5%, zu verzeichnen ist als im vergleichbaren Fall ohne diese Lernstrategie (WP_ANF50). Grund dafür ist die um ein Viertel kleinere Menge von Trainingsmails, die sich nach dem ersten Zyklus ergibt. Nach 86 Trainingsmails enthält das Lexikon üblicherweise 1.115 Wörter, während die hier vorliegenden 66 Trainingsmuster in nur 906 Wörtern resultieren. Dadurch weist auch das Netzwerk weniger Eingabeelemente auf, was die Klassifizierung der 258 Testmails erschwert. Die Differenz in den Testergebnissen gleicht sich innerhalb der 11 Trainingszyklen etwas an, trotzdem bleibt die Version des ACEPS mit *constructive learning* mit 64,0% unter der erreichten Bestleistung von

72,9% (ZÄHL_ANF50). Der Vergleich der Test- und Trainingsergebnisse mit und ohne *constructive learning* findet sich in den Diagrammen 6.9 und 6.10.

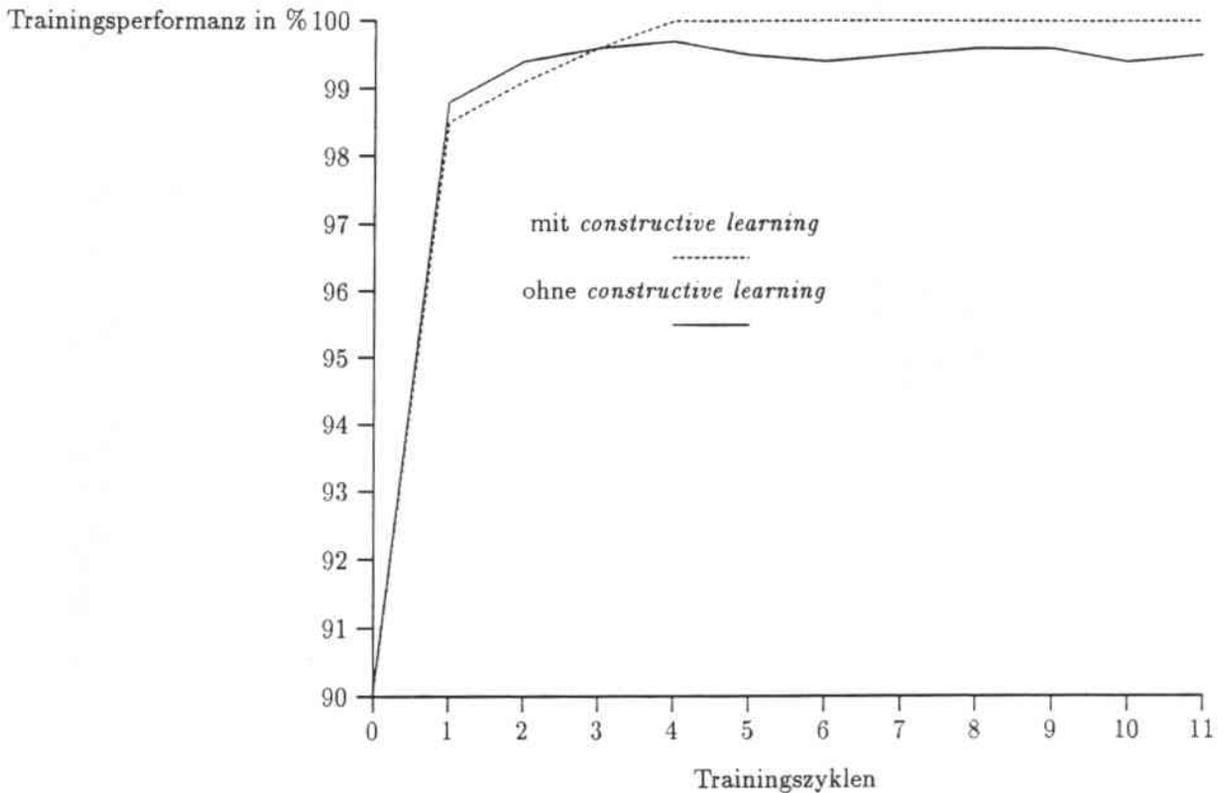


Abbildung 6.9: Vergleich der Trainingsergebnisse mit und ohne *constructive learning*

6.2.6 Manipulation von Verbindungen

Nach der Lernstrategie des *constructive learning* wird in diesem Abschnitt ein weiteres Experiment vorgestellt, das die Strukturen des Netzwerkes während der Lernphase verändert. Wieder wird nur der Anfang einer Mail sowie die Häufigkeit des Auftretens eines Wortes berücksichtigt und das Netz durch benachbarte Wortpaare falsch klassifizierter bereits gelernter Trainingsmuster erweitert. Beim Training des Netzwerkes werden bei dieser Methode alle für die Menge der Trainingsmuster vorgesehenen Mails auch tatsächlich in diese aufgenommen. Wird eine gelernte Mail beim Test der Trainingsdaten falsch klassifiziert, werden nicht nur alle benachbarten Wortpaare ins Netzwerk aufgenommen, sondern auch die Verbindungen zu den Wörtern, die eine falsche Klassifikation verursacht haben, geschwächt. Damit findet eine Korrektur der Verbindungsgewichte des ACEPS-Netzwerkes statt.

Ergebnis dieses Experiments (WEIGHT_ANF50) ist wiederum eine Verbesserung der Trainingsleistung, nicht aber der Performanz auf den Testdaten. Gegenüber dem vergleichbaren Experiment ohne Manipulation der Verbindungen (WP_ANF50) erhöht sich die Genauigkeitsrate beim Training von 99,5% auf 99,8%. Die bisher erreichte Bestleistung auf den Testdaten von 72,9% bleibt unverändert. Die genauen Simulationsergebnisse sind in Tabelle 6.18 im vorigen Abschnitt mit aufgeführt.

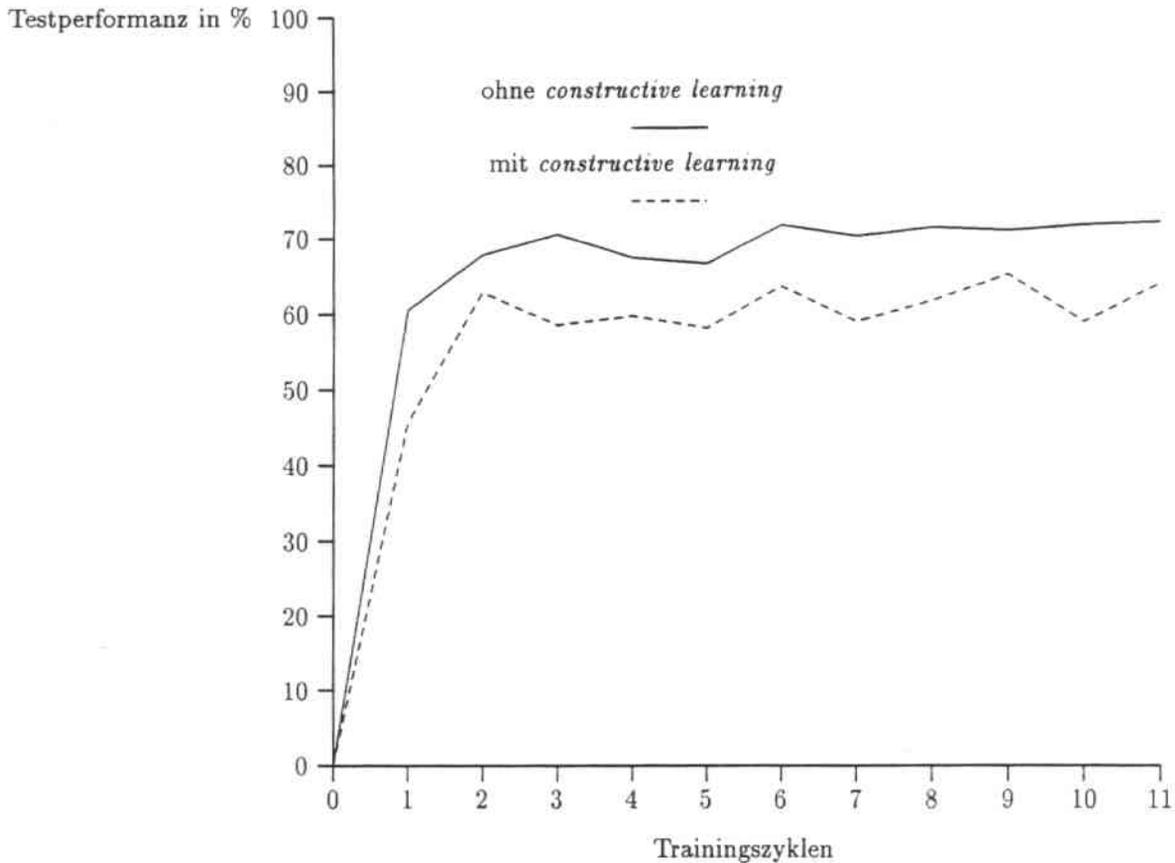


Abbildung 6.10: Vergleich der Testergebnisse mit und ohne *constructive learning*

6.2.7 Reduktion auf lexikalische Wörter

Die in den beiden vorigen Abschnitten ausgeführten Experimente erbrachten eine Verbesserung der Trainingsleistung, haben aber nichts zur Generalisierungsfähigkeit des Netzes beigetragen. Immer noch steht als oberstes Ziel aller Versuche im Vordergrund, die Anzahl der Eingabeelemente des Netzwerkes zu verkleinern und damit die Leistungsfähigkeit des Netzes zu erhöhen.

Bisher wurden im Lexikon des ACEPS Wörter aller Art aufgenommen. Dabei handelt es sich nicht immer nur um Wörter, die auch in einem Wörterbuch zu finden sind. Oft werden per Email z.B. Programmtexte verschickt, in denen sich Zeichenfolgen befinden können, die eigentlich nicht mehr als Wörter zu bezeichnen sind. Im hier beschriebenen Experiment (LEXIKAL) werden diese Zeichenfolgen ausgefiltert und nur Wörter zur Aufnahme ins Lexikon zugelassen, die aus den Buchstaben A bis Z bzw. a bis z bestehen, also auch in einem Duden zu finden sind.

Erstaunlicherweise verbessert sich die Leistung des Systems auf den Testdaten nicht, sondern ist mit 46,9% sogar unwesentlich schlechter als bei Zulassung aller Zeichenfolgen. Zwar nimmt der Wortschatz des Systems um fast 40% auf 16.394 Wörter ab, jedoch gehen einige wichtige Verbindungen, die für die richtige Klassifikation einiger Mails gesorgt haben, verloren. Besonders deutlich wird dies beim Abfall der Trainingsleistung des Systems: von 94,2% nach dem ersten Training sinkt diese bis auf 69,6% nach 11 Zyklen, der

schlechtesten Performanz auf den Trainingsmustern, die bisher überhaupt zu verzeichnen ist.

Dieses Experiment zeigt deutlich, daß es nicht ausreicht, beliebige Zeichenfolgen bei der Klassifikation einer Nachricht nicht zu berücksichtigen. Dies liegt in hohem Maße daran, daß der im ACEPS verwendete Trainingsalgorithmus gerade selten vorkommenden Wörtern eine hohe Gewichtung zuordnet. Dadurch ist eine richtige Klassifikation gelernter Emails gerade durch solche fast einmalig auftretenden Wörter gewährleistet. Diese Tatsache trägt sicherlich nicht zur Generalisierungsfähigkeit des ACEPS bei, sorgt aber für größere Stabilität innerhalb der Trainingsmuster. Durch das Ausfiltern von Wörtern, die Sonderzeichen bzw. Zahlen enthalten und somit nicht nur aus den Buchstaben des Alphabets bestehen, werden viele selten vorkommende Wörter nicht in das Lexikon des ACEPS und auch nicht als Eingabeelemente in das Netzwerk aufgenommen. Beispiele nicht lexikalischer Wörter finden sich in allen der hier verwendeten fünf Email-Kategorien. Besonders in den Themengebieten „Konferenzankündigungen“, „Stellenangebote“ und „Veröffentlichungen“ werden diesen nicht lexikalischen Wörtern hohe Gewichtungen zugeordnet. In der Klasse der „Konferenzankündigungen“ handelt es sich vor allem um die Namen von Konferenzen (*icnn'91, ijcn'91, nips*92*), Uhrzeiten (*12:00, 8:00*) und typische Begriffe wie *co-chairs, camera-ready, non-members, pre-registration, on-campus* und *tutorial/workshop*. In den „Stellenangeboten“ finden sich Ausdrücke wie *post-doctoral, state-funded, opportunity/affirmative, non-canadian, nominees/applicants* und *tenure-track*. Schließlich kommen bei den „Veröffentlichungen“ Formulierungen wie *local-file, remote-file, pub/neuroprose* und *editor-in-charge* sowie die Namen der veröffentlichten Artikel mit dem Suffix *-.ps* vor.

Tabelle 6.19 vergleicht die Ergebnisse dieses Versuches mit ähnlichen Tests im nächsten Abschnitt, die ebenfalls Experimente mit dem Wortschatz des ACEPS betreffen.

6.2.8 Reduktion auf Morpheme

Die bisherigen Experimente zur Reduktion des Wortschatzes des ACEPS haben auf Methoden zurückgegriffen, die sich weniger an der Bedeutung der Wörter an sich als an bestimmten syntaktischen Regeln orientieren. In diesem Abschnitt wird nun die semantische Komponente eines Wortes im Mittelpunkt stehen. Dazu wird das bereits in Kapitel 4 vorgestellte LA-MORPH System von Roland Hausser verwendet.

Mit Hilfe dieses Systems werden alle in den Trainingsmails vorkommenden englischen Wörter auf ihre Morpheme reduziert bevor sie ins Lexikon aufgenommen werden. Die Systemarchitektur des ACEPS in Bild 5.1 wird dann neben Lexikon, Netzwerk und LNW durch eine weitere Komponente, das LA-MORPH System, auf derselben Ebene ergänzt.

Das LA-MORPH System wurde in drei Simulationen des ACEPS verwendet. Im ersten Versuch (MORPH) werden nur die Morpheme von Wörtern berücksichtigt, die im Wörterbuch des LA-MORPH Systems enthalten sind. Bei einem Wortschatz von 3.486 Wörtern ergeben sich dabei schlechte Ergebnisse von 36,0% auf den Testdaten und 48,1% auf den Trainingsdaten. Wie schon bei der Reduktion des Sprachmodells des ACEPS auf Wörter der englischen Sprache sind hier Wörter nicht ins Netz aufgenommen worden, die wesentlich zur Klassifikation der Test- und vor allem der Trainingsdaten beitragen. Dabei

handelt es sich wie schon im letzten Abschnitt um Wörter, die Sonderzeichen beinhalten wie *remote-file*, *local-file*, *pub/neuroprose* und *editor-in-charge* bei den „Veröffentlichungen“ oder Konferenznamen wie *ijcnn'91* und *nips*92* bei den „Konferenzankündigungen“. Vor allem aber fehlen Eigennamen wie *darpa*, *icnn* oder *nips* sowie Fachausdrücke, die nicht im LA-MORPH Wörterbuch enthalten sind. Beispiele dafür sind *overregularization*, *newsgroup*, *hardcopies*, *forthcoming*, *reprints* und *postscript* bei den „Veröffentlichungen“, aber auch fachunspezifischere Formulierungen wie *subscribe* und *unsubscribe* aus dem Gebiet „Unwichtiges“.

Daher werden in einem zweiten Experiment (MORPH*) außer den Morphemen von Wörtern des LA-MORPH Wörterbuchs auch alle Zeichenfolgen, die dort nicht gefunden werden und dementsprechend nicht auf ihre Morpheme reduziert werden können, im Wortschatz des ACEPS zugelassen. Dies führt zu einer Größe des Lexikons von 23.475, einer beinahe unwesentlichen Reduktion von 10% zum ursprünglichen Wortschatz des ACEPS. Abermals sind Test- und Trainingsergebnisse schlechter als bei Berücksichtigung aller Wörter einer Mail. Jedoch hat sich im Vergleich zum vorhergehenden Versuch mit der Zulassung einer größeren Menge von Wörtern auch die Leistung des ACEPS wieder erheblich verbessert. Eine Aufstellung der Ergebnisse dieser Tests mit der Größe des Wortschatzes findet sich in Tabelle 6.19.

Beschreibung	Anzahl Wörter	Reduktion des Wortschatzes	Anzahl Verbindungen	Trainingsleistung	Testleistung
ADPT - Einfache Klassifikation	26.470	—	132.350	76,7%	47,3%
LEXIKAL - Lexikalische Wörter	16.394	38,0%	81.970	69,6%	46,9%
MORPH - Morpheme	3.486	86,8%	17.430	48,1%	36,0%
MORPH* - Morpheme und Zeichenfolgen	23.754	10,3%	118.770	73,1%	45,7%

Tabelle 6.19: Simulationsergebnisse der Experimente mit dem Wortschatz des ACEPS

In den zwei vorhergehenden Experimenten werden die beschriebenen Filtermechanismen unter Zuhilfenahme von LA-MORPH auf den gesamten Text der Email angewendet. Die bislang besten Simulationsergebnisse wurden jedoch bei ausschließlicher Berücksichtigung der Mailanfänge und der Hinzunahme von Wortpaaren erreicht. Aufgrund dieser Tatsache wird der im folgenden beschriebene Versuch (MORPH*_WP) ausgeführt. Es werden nur die ersten 50 Worte einer Email berücksichtigt. Aus diesen 50 Worten werden alle Morpheme und sämtliche Zeichenfolgen, aus denen keine Morpheme gebildet werden können, ins Lexikon aufgenommen. Zur Steigerung der Lernfähigkeit des Netzes werden außerdem noch Wortpaare im Netzwerk zugelassen. Durch die Verwendung des LA-MORPH Systems verringert sich bei 946 Trainingsmails ein vom System gebildeter Wortschatz von 6.428 Wörter auf 5.694 — eine Reduktion um immerhin 11,4%. Dafür werden jedoch mehr Trainingsmails auch nach Aufnahme ihrer Strukturen ins ACEPS falsch klassifiziert,

Beschreibung	Anzahl Wörter	Reduktion des Wortschatzes	Anzahl Wortpaare	Anzahl Verbindungen	Trainingsleistung	Testleistung
WP_ANF50 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare	6.428	—	2166	42.970	99,5%	72,1%
MORPH*_WP - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - Morpheme und Zeichenfolgen	5.694	11,4%	2.369	40.315	99,3%	68,6%

Tabelle 6.20: Experimente mit dem Wortschatz des ACEPS und Wortpaaren

so daß mehr Wortpaare ins Netz integriert werden müssen und die Zahl der Verbindungen sich nur um 6% reduziert. Im Vergleich zum entsprechenden Experiment ohne das LA-MORPH System (WP_ANF50) wird damit nach 11 Trainingszyklen eine annähernd identische Leistung von 99,3% erreicht. Die Genauigkeit, mit der das neue ACEPS die Testdaten klassifiziert, liegt mit 68,6% jedoch 4,3% unter der Bestleistung. Tabelle 6.20 zeigt die genauen Parameter der beiden ACEPS Netzwerke.

6.2.9 Rauschen

Eine übliche Methode bei der Erforschung neuronaler Netze ist die Modifikation der vorhandenen Trainingsmuster durch Rauschen (*noise*), was implizit eine Glättung der Parameter erzeugt. Bei einer sehr geringen Anzahl von Trainingsmustern dient dieses Verfahren dazu, eine größere Anzahl solcher Muster zu schaffen. Diese Vergrößerung der Trainingsmenge ist gerade im Falle des ACEPS eine große Erleichterung, wo die Vorklassifikation von Nachrichten sehr viel Zeit in Anspruch nimmt¹¹. Zum anderen wird dadurch eine gewisse Glättung der Parameter des Netzwerkes geschaffen, da die Trainingsmuster verzerrt werden.

Bei der Anwendung von Rauschen auf die Trainingsmails des ACEPS werden folgende Fälle analysiert:

1. In den Emails wird jedes 10. Wort durch ein Dummy-Wort ersetzt (RAUSCH_DUMMY).
2. Jedes 10. Wort wird durch ein mit Zufallsgenerator erzeugtes Wort aus dem Lexikon ersetzt (RAUSCH_RAND1).

¹¹Zeitmessungen haben ergeben, daß der Mensch ca. 5 Stunden für die Klassifikation der 258 Mails großen Testmenge benötigt. Das bedeutet eine Vorklassifikationszeit von 2,3 Arbeitstagen a 8 Stunden für eine Menge von 946 Trainingsmustern.

3. 10% der Wörter einer Email werden durch andere Wörter ersetzt. Durch Zufalls-generator werden die Wörter der Emails ermittelt, die ersetzt werden sollen und durch ein, ebenfalls mit dem Zufallsgenerator ermitteltes, Wort aus dem Lexikon substituiert (RAUSCH_RAND2).

Durch die Verzerrung der Trainingsmuster ergibt sich jeweils die doppelte Anzahl von Trainingsmails — vorausgesetzt eine Email war nicht ohnehin schon kürzer als zehn Wörter, dann verändert auch das Rauschen in den ersten beiden der drei realisierten Fälle nichts.

In jedem der drei Experimente wird die Eingabe in das ACEPS auf 50 Worte beschränkt und die Worthäufigkeit berücksichtigt. Zusätzlich werden Wortpaare als Elemente in eine Zwischenschicht des Netzwerkes aufgenommen. Bei den beiden zuerst ausgeführten Versuchen (RAUSCH_DUMMY und RAUSCH_RAND1) ergibt sich keine Verbesserung der Testleistung des ACEPS, die Performanz bleibt mit 70,9% und 71,7% annähernd gleich. Die Klassifikation der Trainingsdaten verbessert sich geringfügig auf 99,6%. Beim letzten Experiment (RAUSCH_RAND2) werden sowohl Test- als auch Trainingsleistung etwas schlechter. Die genauen Ergebnisse der drei durchgeführten Simulationen können aus Tabelle 6.21 entnommen werden.

Beschreibung	Anzahl Wörter	Anzahl Verbindungen	Trainingsleistung	Testleistung
RAUSCH_DUMMY - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - <i>Dummy</i> -Wort	6.428	42.970	99,6%	70,9%
RAUSCH_RAND1 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - Zufallswort	6.428	42.970	99,6%	71,7%
RAUSCH_RAND2 - Beschränkung auf 50 Worte - Worthäufigkeit - Wortpaare - 10% Zufallsworte	6.428	42.970	99,3%	65,9%

Tabelle 6.21: Simulationsergebnisse bei Anwendung von Rauschen

6.2.10 Glättung der Parameter (Smoothing)

Auch im Falle des ACEPS ergibt sich das übliche Problem eines Sprachmodells, das auf stochastischen Verfahren beruht. Das ACEPS soll möglichst innerhalb kürzester Zeit sinnvoll anwendbar sein. Der Benutzer, der sich sein Klassifikations- bzw. Priorisierungsschema definiert hat, will nicht noch einige Zeit abwarten bis er eine große

Menge Emails vorklassifiziert hat, sondern die Vorteile des ACEPS sofort in Anspruch nehmen. Gleichzeitig schreibt die Anwendung eine unbeschränkte Eingabe von Text vor, das zulässige Vokabular läßt sich also nicht eingrenzen. Damit ergibt sich zwangsläufig ein Mißverhältnis zwischen der Anzahl zur Verfügung stehender Trainingsemails und der uneingeschränkt wachsenden Menge von Modellparametern. Auch beim ACEPS läßt sich aus diesem Grund eine mangelhafte Generalisierungsfähigkeit erkennen. Durch seinen großen Wortschatz erzielt das System somit eine schlechtere Performanz als es der Fall wäre, wenn es sich um ein System mit eingeschränktem Vokabular handeln würde. Das Netzwerk ist zwar in der Lage, sich an einmal gelernte Muster zu erinnern und diese auch bei Erwerb neuen Wissens weiter richtig zu klassifizieren. Jedoch ist keine wesentliche Leistungssteigerung zu verzeichnen je länger das ACEPS trainiert wird und je mehr Wissen es eigentlich besitzen müßte. Dies beweist eine Analyse der Performanzmessungen auf der Menge der Testdaten. Bild 6.11 zeigt ein Diagramm der Testergebnisse des Experiments WP_ANF50 nach den 11 Lernzyklen des ACEPS. Man sieht, daß die Performanz sich

Performanzmessungen

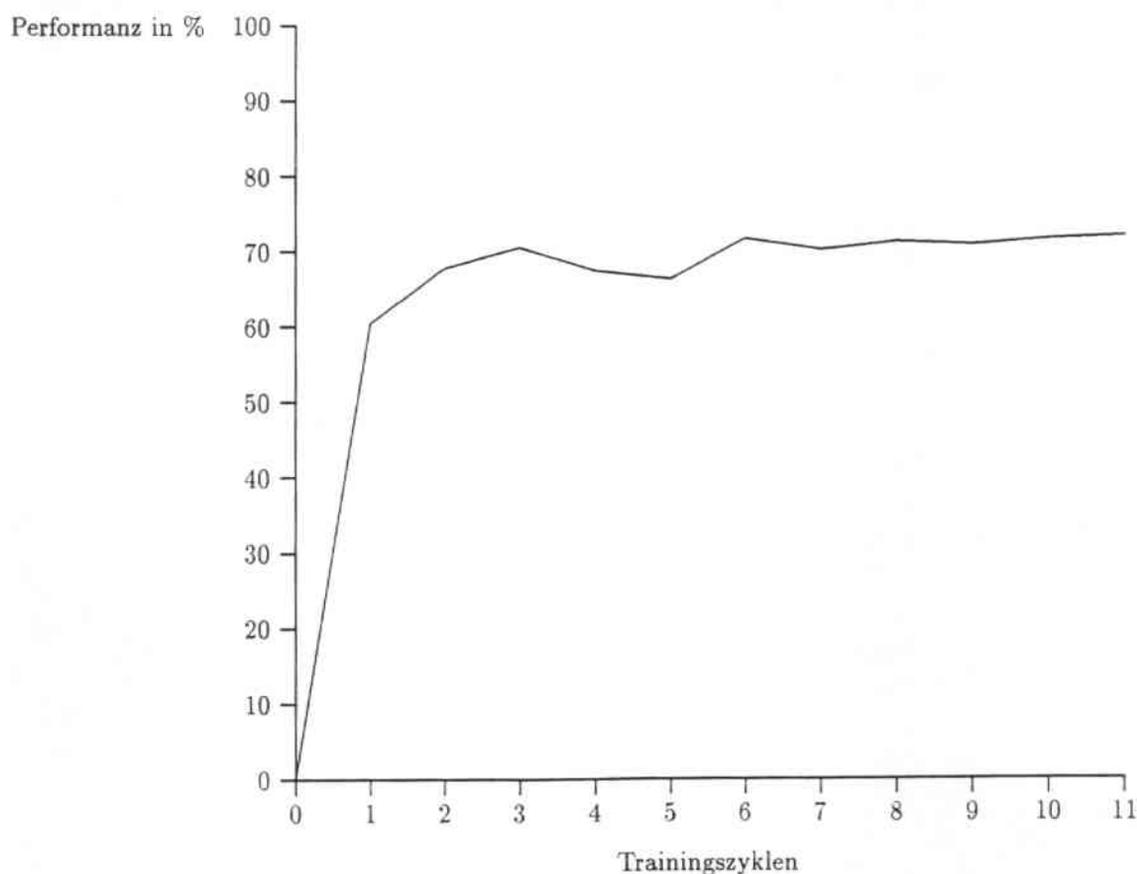


Abbildung 6.11: Testperformanzmessungen des Experiments WP_ANF50

zwar von der ersten zur letzten Trainingseinheit verbessert hat, jedoch ist die endgültige Leistung von 72,1% bereits nach 6 Trainingszyklen erreicht. Zu diesem Zeitpunkt sind erst 516 Trainingsmuster gelernt worden. Die restlichen 430 Trainingsmails verschlechtern die Performanz erst einmal, um dann auch nur eine Steigerung von 0,5% zu erreichen.

Ein solches „Overtraining“ verlangt geradezu nach einer Glättung der Parameter durch das schon in Abschnitt 4.4 erwähnte Cooccurrence Smoothing. Die hier durchgeführte Glättung und die benutzte Formel wird im folgenden theoretisch erläutert und die erworbenen Ergebnisse werden vorgestellt.

Sei v_n eine Zufallsvariable, die ein Wort v_n an der Position n des Lexikons des ACEPS angibt. Die Wahrscheinlichkeit des Vorkommens dieses Wortes innerhalb des geschriebenen Textes aller Emails ergibt sich dann wie schon im Trainingsalgorithmus in Abschnitt 5.3.3 beschrieben. Die Wahrscheinlichkeiten aller 6.428 Wörter, die sich ja aus einer nur 946 Muster großen Trainingsmenge ergeben, sollen nun geglättet werden. Eine Möglichkeit, zuverlässigere Schätzungen der bedingten Wahrscheinlichkeiten eines Wortes unter der Voraussetzung einer bestimmten Ausgabeaktion zu erhalten, ist das Ausnutzen der Gleichartigkeit von Wörtern. Die bedingte Wahrscheinlichkeit eines Wortes v_n ergibt sich dann aus der Wahrscheinlichkeit, mit der das Wort v_n des Wortpaares (v_n, v_m) dafür sorgt, daß dieselbe semantische Aktion vom Netz ausgeführt wird, die schon aus dem Wort v_m resultierte.

Die Definition von Cooccurrence Smoothing beruht auf der bedingten Cooccurrence-Wahrscheinlichkeit $P_c(v_n|v_m)$ zweier Wörter v_n und v_m . Die Cooccurrence-Wahrscheinlichkeit ist dabei folgendermaßen definiert:

$$P_c(v_n|v_m) = \frac{\sum_{k=1}^5 P(c_k|v_n) \cdot P(c_k|v_m) \cdot P(c_k)}{\sum_{p=1}^{\#words} \sum_{k=1}^5 P(c_k|v_p) \cdot P(c_k|v_m) \cdot P(c_k)}$$

Sind die bedingten Wahrscheinlichkeiten $P(c_k|v_p)$ aller Wörter p untertrainiert, was oft der Fall ist, sind die Wahrscheinlichkeitswerte sehr unterschiedlich und viele Wahrscheinlichkeiten sind gleich Null. Verständlicherweise führt eine solche Verteilung zu schlechten Klassifikationsresultaten des Netzes. Daher wird statt der üblichen bedingten Wahrscheinlichkeit die geglättete Wahrscheinlichkeit P_s benutzt:

$$P_s(c_k|v_n) = \sum_{i=1}^{\#words} P_c(v_n|v_i) \cdot P(c_k|v_n).$$

Beschreibung	Anzahl Wörter	Anzahl Verbindungen	Trainingsleistung	Testleistung
SMOOTH - Beschränkung auf 50 Worte - Worthäufigkeit - <i>cooccurrence smoothing</i>	6.428	32.140	88,9%	62,4%

Tabelle 6.22: Simulationsergebnisse nach Einführung von *cooccurrence smoothing*

Die Auswertung dieses Experiments (SMOOTH) (siehe Tabelle 6.22) ergibt eine Leistung von 88,9% auf den Trainingsdaten und 62,4% auf den Testdaten. Zwar werden durch dieses Verfahren die Werte der Verbindungsgewichte geglättet, jedoch gleichen sich die Wahrscheinlichkeiten der einzelnen Klassen damit so weit aneinander an, daß die richtige Klassifikation einzelner Mails erschwert wird.

6.2.11 Verwendung von Backpropagation als Trainingsalgorithmus

Der in allen vorangehenden Simulationen benutzte Trainingsalgorithmus beruht auf der Schätzung von Wahrscheinlichkeiten. Die dahinterstehende Theorie ist in Kapitel 4 vorgestellt worden. Dort wurde auch bereits der allgemeine Algorithmus einer Backpropagation-Trainingsprozedur eingeführt.

Gerade die große Diskrepanz zwischen der nahezu perfekten Trainingsleistung und der viel schwächeren Leistung auf den Testdaten läßt darauf schließen, daß durch weiteres Anpassen der Verbindungsgewichte eine Leistungssteigerung möglich ist. Deshalb wird das ACEPS hier nun durch eine andere Art des Lernens trainiert. Als Trainingsalgorithmus wird der bereits beschriebene Backpropagation-Algorithmus verwendet. Das erste Experiment (BP_RAND), das mit Backpropagation durchgeführt wird, initialisiert alle Verbindungsgewichte mit Zufallswerten zwischen +0,3 und -0,3. Nach 26 Epochen bzw. Iterationen Backpropagation-Training erreicht das ACEPS seine bisherige Höchstleistung, die 77,9% Genauigkeit auf den Testdaten beträgt. Nach dieser Anzahl von Zyklen ist auch die höchste Genauigkeit von 99,9% bei der Klassifikation der Trainingsmuster erzielt. Auch nach mehreren 100 Epochen werden beide Ergebnisse nicht verbessert, die Testperformanz konvergiert gegen 77,5%, die Trainingsleistung bleibt bei 99,9%.

Nachteil des Trainings durch Backpropagation ist die Dauer des Lernvorgangs. Dies wird besonders deutlich, berücksichtigt man Emails in ihrer vollen Länge (BP). Mit Hilfe des Backpropagation-Algorithmus wird nach 26 Iterationen zwar 75,6% Genauigkeit bei den Testdaten und 96,6% bei den Trainingsdaten erreicht. Dies ist die höchste Leistung, die bisher bei einer Anzahl von 26.470 Netzelementen erreicht wurde — die beste Leistung bis zu diesem Zeitpunkt bei Berücksichtigung der Nachricht in voller Länge war die Realisierung von 3 Teilnetzen mit gleicher Gewichtung und führte zu 49,6%. Diese 26 Epochen nehmen jedoch 143 CPU-Stunden in Anspruch und dauern je nach Rechnerauslastung 5 bis 6 Tage. In derselben Zeit sind 1.226 Iterationen bei einer Kürzung der Mail auf die ersten 50 Worte möglich (BP_RAND), wobei kürzere Mails auch mit besserer Performanz klassifiziert werden.

Eine weitere praktikable Möglichkeit, die Trainingszeit zu verkürzen, ist eine Initialisierung der Verbindungsgewichte mit Hilfe des bisher benutzten adaptiven Trainingsalgorithmus. Durch die Schätzung von Wahrscheinlichkeiten ist bereits eine bessere Voreinstellung der Gewichte erreicht als durch zufällig ermittelte Werte und eine erste Klassifikation der Testdaten kann schon bei einer Genauigkeit von 72,9% anstelle z.B. 14,0% beginnen. Die weitere Feineinstellung der Parameter des ACEPS-Netzwerkes erfolgt unter Zuhilfenahme von Backpropagation-Training.

Beschreibung	Anzahl Wörter	Anzahl Verbindungen	Anzahl Epochen	Trainingsleistung	Testleistung
BP - Backpropagation - Initialisierung durch Zufallszahlen	26.470	132.350	26	96,6%	75,6%
BP_RAND - Beschränkung auf 50 Worte - Backpropagation - Initialisierung durch Zufallszahlen	6.428	32.140	26	99,9%	77,9%
			91	99,9%	77,5%
BP_ADPT - Beschränkung auf 50 Worte - Worthäufigkeit - Backpropagation - Initialisierung durch Schätzung von Wahrscheinlichkeiten	6.428	32.140	9	99,5%	79,1%
			72	100%	76,0%
BP_WD1 - Beschränkung auf 50 Worte - Worthäufigkeit - Backpropagation - Initialisierung durch Schätzung von Wahrscheinlichkeiten - Gewichtsverminderung (Variante 1)	6.428	32.140	4	99,1%	77,9%%
			21	100%	77,9%
BP_WD2 - Beschränkung auf 50 Worte - Worthäufigkeit - Backpropagation - Initialisierung durch Schätzung von Wahrscheinlichkeiten - Gewichtsverminderung (Variante 2)	6.428	32.140	5	95,4%	77,5%
			11	95,0%	77,1%

Tabelle 6.23: Ergebnisse der Simulationen mit Backpropagation durch das ACEPS

Tatsächlich sorgt dieses Experiment (BP_ADPT) dafür, daß nach bereits 9 Epochen Backpropagation-Training der bisherige Höchststand der Leistung auf den Testdaten überhaupt erreicht ist: 79,1%. Die zu diesem Zeitpunkt erworbene Genauigkeitsrate auf der Trainingsmenge beträgt 99,5%. In den nachfolgenden Epochen ist keine Steigerung der Testleistung zu verzeichnen, sie pendelt sich auf 76,0% ein. Die Trainingsleistung nimmt jedoch nach 72 Iterationen auf 100% zu und diese perfekte Leistung verliert das System nicht mehr.

Durch das iterative zyklische Lernen des Backpropagation-Algorithmus wird die Stabilität des ACEPS erheblich erhöht. Leider ist die erwünschte Generalisierungsfähigkeit auch mit diesem Mittel nicht erreicht.

Um zu vermeiden, daß die Gewichte durch viele hundert Iterationen beim Backpropagation immer größer werden, wurde eine weitere Version des ACEPS als Simulation gesteuert, die durch *weight decay* (siehe z.B. [Moo91]) bei jeder Gewichtsveränderung durch eine Verminderung dafür sorgt, daß die Größe der Verbindungsgewichte möglichst klein bleibt. Ziel dieser Strategie ist es, die Generalisierungsfähigkeit des Netzes zu verbessern.

Tests wurden mit zwei verschiedenen Verminderungsfaktoren durchgeführt. Bei beiden Experimenten (BP_WD1 und BP_WD2) werden jeweils nur die ersten 50 Worte einer Email als Eingabe verwendet und dabei auch die Häufigkeit des Auftretens eines Wortes berücksichtigt. Die Initialisierung der Netzgewichte erfolgt durch Schätzung von Wahrscheinlichkeiten mit Hilfe des adaptiven Trainingsalgorithmus des ACEPS. Danach erfolgt die Feineinstellung der Parameter durch Backpropagation-Training. Die erreichte Leistung resultiert in einem Ergebnis, das mit dem einer zufälligen Gewichtsinitialisierung vergleichbar ist. Die erste Variante (BP_WD1) des *weight decay* erreicht nach 4 Epochen dieselbe Leistung wie schon BP_RAND: 77,9%. Die Trainingsleistung ist zu diesem Zeitpunkt 99,1%. Damit ist zwar keine Leistungssteigerung zu verzeichnen, aber der Lernvorgang wird erheblich beschleunigt. Eine perfekte Trainingsleistung wird statt nach 72 Iterationen bereits nach 21 Epochen erreicht. Schließlich pendelt sich das Ergebnis nach 41 Epochen auf 78,3% Testleistung ein, die Trainingsleistung bleibt ab der 21. Epoche stabil bei 100%. Der in BP_WD2 gewählte Verminderungsfaktor führt zu etwas schlechteren Ergebnissen.

Alle Simulationsergebnisse der mit dem Backpropagation-Algorithmus trainierten Netze sind noch einmal in Tabelle 6.23 dargestellt.

6.3 Zusammenfassung der Ergebnisse

Um eine übersichtlichere Darstellung der durchgeführten Simulationen zu gewährleisten, werden in Tabelle 6.24 noch einmal alle erreichten Ergebnisse zusammengefaßt. Dabei werden Name des Experiments genannt sowie der Abschnitt, in dem eine ausführliche Beschreibung des durchgeführten Versuchs zu finden ist. Neben den Test- und Trainingsergebnissen ist außerdem die Zahl der vorkommenden Wörter und Verbindungen in der Tabelle festgehalten.

Abbildung 6.12 stellt in einem Diagramm die wichtigsten Test- und Trainingsergebnisse des ACEPS unter Benutzung verschiedener Netzwerkarchitekturen graphisch dar. Als re-

Bezeichnung des Experiments	Abschnitt	Anzahl Wörter	Anzahl Verbindungen	Anzahl Epochen	Trainings- leistung	Test- leistung
MENSCH	6.1.3	?	$\approx 10^{14}$?	?	79.4%
ADPT	6.2.1	26.470	132.350	1	76,7%	47,3%
ZÄHL	6.2.1	26.470	132.350	1	76,1%	46,9%
ADPT_ANF50	6.2.2	6.428	32.140	1	96,0%	72,1%
ZÄHL_ANF50	6.2.2	6.428	32.140	1	96,6%	72,9%
ADPT_WORT5	6.2.2	17.541	87.705	1	63,0%	41,9%
BIGNET	6.2.3	26.470	132.350	1	78,6%	49,6%
BIGNET_LK	6.2.3	26.470	132.350	1	78,6%	48,8%
BIGNET_ANF50	6.2.3	6.428	32.140	1	97,6%	67,4%
WP_ANF50	6.2.4	6.428	42.970	1	99,5%	72,1
CLEARN_ANF50	6.2.5	3.669	19.560	1	100%	64,0%
WEIGHT_ANF50	6.2.6	6.428	42.940	1	99,8%	72,9%
LEXIKAL	6.2.7	16.394	81.970	1	69,6%	46,9%
MORPH	6.2.8	3.486	17.430	1	48,1%	36,0%
MORPH*	6.2.8	23.754	118.770	1	73,1%	45,7%
MORPH*_WP	6.2.8	5.694	40.315	1	99,3%	68,6%
RAUSCH_DUMMY	6.2.9	6.428	42.970	1	99,6%	70,9%
RAUSCH_RAND1	6.2.9	6.428	42.970	1	99,6%	71,7%
RAUSCH_RAND2	6.2.9	6.428	42.970	1	99,3%	65,9%
SMOOTH	6.2.10	6.428	32.140	1	88,9%	62,4%
BP	6.2.11	26.470	132.350	26	96,6%	75,6%
BP_RAND	6.2.11	6.428	32.140	26	99,9%	77,9%
BP_ADPT	6.2.11	6.428	32.140	9	99,5%	79,1%
BP_WD1	6.2.11	6.428	32.140	21	100%	77,9%
BP_WD2	6.2.11	6.428	32.140	5	95,4%	77,5%

Tabelle 6.24: Zusammenfassung der Simulationsergebnisse

levante Versuche werden lediglich diejenigen Tests aufgeführt, die entscheidende Veränderungen gegenüber bisher durchgeführten Simulationen bewirkten.

6.4 Fehleranalyse

Betrachtet man die Simulationsergebnisse aller Experimente, die mit dem ACEPS durchgeführt wurden, zeigt sich, daß das System in der Lage ist, Trainingsmuster mit bis zu

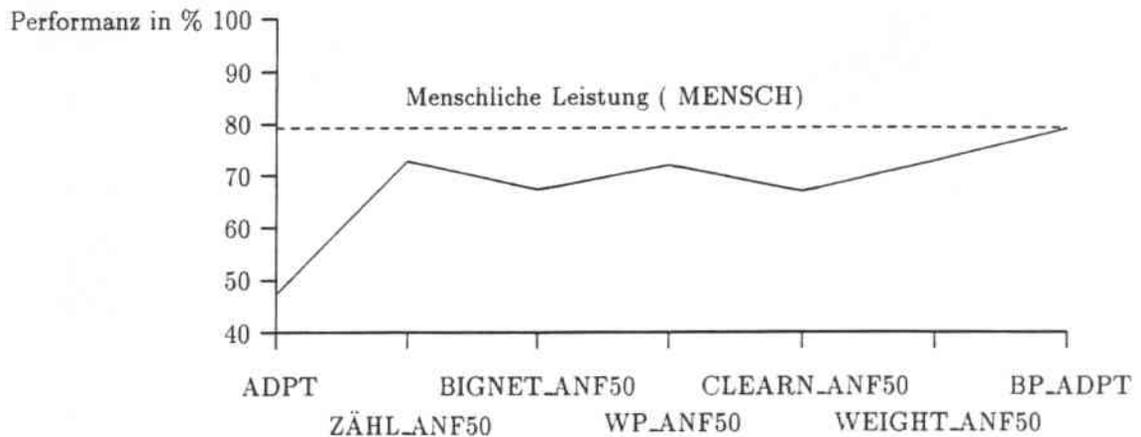


Abbildung 6.12: Ergebnisse unter Benutzung verschiedener Architekturen

100%iger Genauigkeit zu klassifizieren. Die erreichte Leistung auf der Menge der Testmuster bleibt mit nur 79,1% weit hinter dieser Genauigkeit zurück.

Dabei lassen sich in einer Fehleranalyse zwei Sachverhalte untersuchen:

1. Die Performanz auf den Trainingsdaten
2. Die Performanz auf den Testdaten

Bei der Analyse dieser Beobachtungen geht dieser Abschnitt zunächst auf die erzielten Resultate der Klassifikation der Trainingsmenge ein. Danach findet eine Untersuchung der Performanz auf den Testdaten und eine Analyse der gemachten Fehlklassifikationen statt.

6.4.1 Performanz auf den Trainingsdaten

Zur Analyse der Trainingsleistung des ACEPS wird das Ergebnis des Experiments WP_ANF50 aus Abschnitt 6.2.4 mit einer Trainingsperformanz von 99,5% nach 11 Zyklen ausgewählt. Bei diesem Versuch wird der Eingabetext auf 50 Worte begrenzt und die Worthäufigkeit eines Wortes mit berücksichtigt. Zusätzlich wird das ACEPS-Netzwerk durch eine Zwischenschicht ergänzt, deren Elemente Wortpaare repräsentieren. Dabei wird der Trainingsalgorithmus des ACEPS benutzt, der ausschließlich auf der Schätzung von Wahrscheinlichkeiten beruht.

Nach dem ersten Trainingszyklus wird hier nur eine Mail falsch klassifiziert, was bei einer Gesamtheit von 86 Mails zu einer Genauigkeit von 98,8% führt. Nach 11 Trainingszyklen werden insgesamt 5 Emails falsch eingeordnet. Die Konfusionsmatrizen beider Trainingszyklen sind in den Tabellen 6.25 und 6.26 angegeben.

Bei weiterer Auswertung der gemachten Fehlklassifikationen zeigt sich, daß die schon nach dem ersten Zyklus falsch klassifizierte Mail auch noch nach dem letzten Training falsch klassifiziert wird. Grund dafür ist das gleichverteilte Vorkommen aller Wörter dieser Mail

	Konferenz-ankündigungen	Fachliche Diskussion	Stellenangebote	Unwichtiges	Veröffentlichungen
Konferenz-ankündigungen	20	0	1	0	0
Fachliche Diskussion	0	20	0	0	0
Stellenangebote	0	0	5	0	0
Unwichtiges	0	0	0	20	0
Veröffentlichungen	0	0	0	0	20

Tabelle 6.25: Konfusionsmatrix nach 1. Trainingszyklus

	Konferenz-ankündigungen	Fachliche Diskussion	Stellenangebote	Unwichtiges	Veröffentlichungen
Konferenz-ankündigungen	220	0	0	4	1
Fachliche Diskussion	0	220	0	0	0
Stellenangebote	0	0	66	0	0
Unwichtiges	0	0	0	216	0
Veröffentlichungen	0	0	0	0	219

Tabelle 6.26: Konfusionsmatrix nach 11. Trainingszyklus

in drei der fünf vorgegebenen Klassen. Dies resultiert in identischen Aktivierungen aller drei Klassen. Das System ordnet die Email dann in die Klasse ein, die vom Benutzer beim initialen Aufruf des Systems als erste im Klassifikationsrahmen definiert wurde.

Vier der fünf falsch klassifizierten Emails stammen aus der Klasse „Unwichtiges“. Alle vier werden vom ACEPS fälschlicherweise in „Konferenzankündigungen“ eingeordnet. Bei einer dieser vier Dateien ist die bereits oben erwähnte Gleichaktivierung von drei Klassen der Grund. Die richtige Klasse hat zwar dieselbe Aktivierung wie die zwei „falschen“, jedoch führt die Definitionsreihenfolge des Klassifikationsschemas zu diesem vom ACEPS gemachten Fehler. Die restlichen drei Trainingsmails enthalten Wörter, die deutlich für ihre Einordnung in den Bereich „Konferenzankündigungen“ sprechen. Dabei handelt es sich um Wörter wie *call for papers*, *international joint conference neural networks* und *submissions*¹². Die vom ACEPS ausgeführte Aktion ist also von der Bedeutung der Wörter her gesehen richtig. Betrachtet man jedoch den Kontext, in dem sie stehen, wird aus dem Zusammenhang deutlich, daß eine Einordnung in „Unwichtiges“ gerechtfertigt ist. Eine

¹²Anhang C enthält Wortlisten, die für jede der fünf gewählten Klassen „typische“ Wörter, d.h. Wörter mit hohen Gewichten, auführt.

der Mails besteht nur aus einer „subject“- und „From“-Zeile, der Textkorpus ist völlig leer. Eine Nachricht, die lediglich *call for papers* ... enthält, ist für einen Leser nutzlos. In einer anderen Email wird über die Mailing-Liste nach der Adresse für Konferenzbeiträge gefragt — je nach Gutdünken eines Lesers läßt sich diese Mail in „Unwichtiges“ oder „Konferenzankündigungen“ einordnen.

Bei der letzten der „falsch“ klassifizierten Mails handelt es sich um keine echte Fehlleistung des ACEPS. Vielmehr ist ein Irrtum bei der Vorklassifikation der Trainingsmuster festzustellen. Die Mail enthält die Ankündigung eines *Workshops* über *machine learning* und gehört daher nicht zur Klasse der „Veröffentlichungen“, sondern ist tatsächlich eine „Konferenzankündigung“. Die Zahl falsch klassifizierter Emails reduziert sich dadurch also auf vier¹³. Die fünf falsch klassifizierten Emails sind in Anhang D aufgelistet.

Das ACEPS ist also in der Lage, vier der fünf vorgegebenen Klassen innerhalb der Trainingsmails gut voneinander abzugrenzen. Lediglich die Klasse „Unwichtiges“ bereitet dem System Schwierigkeiten. Die Berücksichtigung der Struktur einer Nachricht, z.B. ein leerer Textkorpus, bzw. weitere Einbeziehung der Bedeutung von Wortfolgen können in zukünftigen ACEPS-Versionen dazu beitragen, die Leistung des Systems weiter zu verbessern.

Bereits durch die Manipulation der Verbindungen (WEIGHT_ANF50) in Abschnitt 6.2.6 wurden nach 11 Trainingszyklen nur noch zwei Emails des Ordners „Unwichtiges“ falsch klassifiziert. Die Strategie des *constructive learning* führt bereits nach 3 Trainingszyklen zu 100%iger Stabilität auf den Trainingsmustern. Die Tatsache, daß nicht alle vorklassifizierten Ein-/Ausgabepaare in die Trainingsmenge aufgenommen werden, bewirkt folgendes: Die Wörter, die in den zwei bisher auch nach 11 Zyklen noch nicht gelernten Emails vorkommen, haben positive Gewichte zu den richtigen Ausgabeelementen und Werte gleich Null zu den übrigen Klassen.

Eine Analyse der Trainingsleistung der ACEPS-Version, bei der eine Kombination von adaptivem und *Backpropagation*-Lernen benutzt wurde (BP_ADPT), findet hier nicht statt. In diesem Experiment resultiert die Lernphase in einer 100%igen Stabilität des ACEPS-Netzwerkes. Durch die Initialisierung der Verbindungsgewichte mit Wahrscheinlichkeiten anstelle zufällig ermittelter Werte wird die Dauer des Lernvorgangs auf wenige Zyklen begrenzt, was einer akzeptablen Länge der Trainingsphase entspricht.

6.4.2 Performanz auf den Testdaten

Zur Beurteilung der Leistungsfähigkeit des ACEPS-Netzes werden ebenfalls die durchgeführten Tests (WP_ANF50) aus Abschnitt 6.2.4 betrachtet. Dabei werden besonders die Leistungstests nach dem 1. und 11. Trainingszyklus berücksichtigt.

Im 1. Test der Testmenge des ACEPS wird mit einer Genauigkeit von 60,5% klassifiziert, was einer Anzahl von Fehlklassifikationen von 102 Emails entspricht. Nach dem letzten Test werden noch 72 Fehler gemacht, dies entspricht einer Genauigkeit von 72,1%. Dabei stellen sich folgende Fragen:

¹³Das Etikett dieser Email wurde bei Erkennen des Fehlers nicht geändert, um weiterhin vergleichbare Ergebnisse aller Simulationen zu gewährleisten, ohne alle Experimente wiederholen zu müssen.

- Wird die Performanz auf der Testmenge besser je mehr das Netz trainiert, je größer das Lexikon und je komplexer das Netzwerk wird?
- Werden Emails, die zu Anfang falsch klassifiziert wurden, „gelernt“ und somit später richtig klassifiziert?

Dabei stellt man fest, daß die Leistungen des Netzes sehr schwankend sind. Eine einmal richtig klassifizierte Email wird nicht notwendigerweise auch im nächsten bzw. allen anderen Zyklen richtig eingeordnet. Es kann vorkommen, daß sich die Gewichte der Verbindungen durch das durchgeführte Training so verändern, daß eine ehemals richtig behandelte Email im darauffolgenden Zyklus wieder falsch klassifiziert wird. Durch erneutes Training kann dieser Fehler im nächsten Zyklus wieder korrigiert werden, jedoch zeigt dieses Verhalten deutlich die Wechselhaftigkeit bei der Klassifikation der Testdaten.

	Konferenz-ankündigungen	Fachliche Diskussion	Stellenangebote	Unwichtiges	Veröffentlichungen
Konferenz-ankündigungen	53	13	8	8	14
Fachliche Diskussion	3	23	0	4	3
Stellenangebote	0	1	7	1	1
Unwichtiges	0	4	1	31	0
Veröffentlichungen	4	19	2	16	42

Tabelle 6.27: Konfusionsmatrix nach dem 1. Test

	Konferenz-ankündigungen	Fachliche Diskussion	Stellenangebote	Unwichtiges	Veröffentlichungen
Konferenz-ankündigungen	55	3	3	3	6
Fachliche Diskussion	0	38	1	12	4
Stellenangebote	0	0	12	0	0
Unwichtiges	0	7	1	35	2
Veröffentlichungen	5	12	1	10	48

Tabelle 6.28: Konfusionsmatrix nach dem 11. Test

Betrachtet man die in Tabelle 6.27 und Tabelle 6.28 dargestellten Konfusionsmatrizen, läßt sich folgendes feststellen: Besonders häufig treten Fehlklassifikationen in der Klasse

„Fachliche Diskussion“ auf. Dort werden beim 1. Test nur 38% aller Emails richtig klassifiziert und auch nach 11 Zyklen steigert sich diese Performanz nur auf 63%. Die „Konferenzankündigungen“ dagegen werden nach dem ersten Test mit 88% Genauigkeit klassifiziert. Diese Leistung steigert sich nach 11 Zyklen auf fast 92%! Die Genauigkeit, mit der das ACEPS nach dem 1. und 11. Test klassifiziert, ist nach Themengebieten unterteilt noch einmal in Tabelle 6.29 dargestellt.

Themen- gebiet	Performanz nach 1. Test	Performanz nach 11. Test
Konferenz- ankündigungen	88%	92%
Fachliche Diskussion	38%	63%
Stellen- angebote	39%	67%
Unwichtiges	52%	58%
Veröffentlichungen	70%	80%
Gesamt	60,5%	72,1%

Tabelle 6.29: Performanz je Themengebiet nach 1. und 11. Test

Der Ordner „Konferenzankündigungen“ ist also mit nur 5 gemachten Fehlern besser von allen anderen Ordnern abgrenzbar als die übrigen Klassen. Dabei ordnet das ACEPS alle falsch klassifizierten Mails in den Ordner „Veröffentlichungen“ ein. Eine genaue Analyse der Aktivierungen der einzelnen Klassen zeigt, daß die „Veröffentlichungen“-Aktivierung nur geringfügig höher ist als die der richtigen Klasse „Konferenzankündigungen“. Dieser Fehler läßt sich vermeiden, führt man eine „Unbekannt“-Klasse als weitere Klassifikationsmöglichkeit ins ACEPS ein. Ist die maximale Aktivierung eines Ordners nicht um einen konstanten Wert größer als die zweithöchste Aktivierung, wird die Email in „Unbekannt“ eingeordnet. Weiß das ACEPS mit hoher Wahrscheinlichkeit, daß es einen Fehler machen wird, ist es für den Benutzer erstrebenswerter, die Klassifikation selbst vorzunehmen als eine falsche Aktion zu korrigieren.

Der Vergleich des 1. und 11. Tests zeigt auch, daß nach 11 Zyklen keinerlei Falschklassifikationen mehr in den Ordner „Stellenangebote“ gemacht werden. Jedoch finden sich immer noch einige Emails der Klasse „Stellenangebote“, die in andere Ordner falsch eingeordnet werden, so daß man auch hier nicht von einer klaren Abgrenzung zu den übrigen Klassen sprechen kann. Insgesamt sind nach 11 Zyklen in allen 5 Klassen Verbesserungen der Performanz festzustellen. Bei den 72 Testmails, die noch immer in falsche Ordner eingeordnet werden, reicht die Aktivierung der schon im Lexikon vorhandenen Wörter nicht aus, um eine richtige Klassifikation zu gewährleisten. Sie enthalten viele Wörter, die nicht im Lexikon vorhanden sind. Dabei handelt es sich jedoch nicht um Wörter, die mit dem Themenbereich des Konnektionismus in Zusammenhang stehen, sondern um Eigennamen und allgemeine selten benutzte Wörter. Damit zeigt sich, daß eine Vergrößerung des Le-

xikons zwar zur Lösung dieses Problems beitragen würde, damit aber auch gleichzeitig die Anzahl der Parameter des Netzwerkes zunimmt. Schon die ganz zu Anfang durchgeführten Experimente des ACEPS, bei denen Nachrichten in voller Länge zugelassen wurden, haben gezeigt, daß mit einem solchen Vorgehen aber noch immer nicht die grundlegende Schwierigkeit beseitigt wird: die mangelnde Generalisierungsfähigkeit des ACEPS.

Um diesen Mangel zu beheben, müssen andere Möglichkeiten gefunden werden — eine bloße Vergrößerung des Lexikons und damit des Netzwerkes reicht dafür nicht aus.

6.4.3 Abschließende Betrachtungen

Bei Vergrößerung der Trainingsmenge des ACEPS und damit einem weiteren Wachstum des Wortschatzes ist nicht auf jeden Fall eine bessere Generalisierungsfähigkeit des ACEPS-Netzwerkes gewährleistet. Das Problem beruht nicht ausschließlich auf dem *Overtraining* der Parameter des Netzes, dies ist schon bei Einführung des Cooccurrence Smoothing deutlich geworden. Deshalb müssen andere Möglichkeiten gefunden werden, um die Performanz des Systems zu verbessern.

Die leichte Leistungssteigerung des ACEPS bei Verwendung von 3 Teilnetzen läßt sich durch eine bessere Kombination der Aktivierungen dieser Netze noch verstärken. Dazu ist jedoch ein Verfahren notwendig, das eine optimale Linearkombination ermittelt.

Neben Gegenstand und Absender einer Nachricht ist auch eine Berücksichtigung des Aufbaus und der Gliederung der Email beim Klassifikationsvorgang des ACEPS möglich. Dazu gehört z.B. die Länge einer Nachricht und ein leerer Textkorpus oder eine fehlende „subject“-Zeile.

Auch der Ansatz des *constructive learning* kann durch Erweiterung auf eine *constructive/destructive learning*-Strategie verbessert werden. Dabei werden nicht nur neue Eingabelemente in das ACEPS-Netzwerk aufgenommen, sondern Wörter können auch wieder aus dem Lexikon gelöscht werden. Dies ist dann sinnvoll, wenn sie nicht maßgeblich an der richtigen Klassifikation einer Email mitwirken bzw. in allen Klassen des ACEPS mit gleicher Wahrscheinlichkeit vorkommen und damit nichts zu einer Klassifikation beitragen.

Die Hinzunahme von Wortpaaren in die Strukturen des ACEPS-Netzwerkes haben nur zu einer Steigerung der Trainingsleistung beigetragen. Eine weitere Berücksichtigung der Reihenfolge von Wörtern, die über Wortpaare hinausgeht, läßt aber auch eine Steigerung des Performanz auf den Testdaten erwarten.

Schließlich ist eine Erhöhung der Leistung des ACEPS auch durch weitere Tests mit verschiedenen Parametern des Backpropagation-Algorithmus denkbar.

Viele dieser Ideen konnten wegen der langen Dauer der Simulationen nicht verwirklicht werden, da ein neuronales Netzwerk eine lange Rückkopplungsphase benötigt bis Ergebnisse vorhanden sind. Aus dem gleichen Grund sind auch viele der Experimente nur mit einer beschränkten Eingabe in das ACEPS durchgeführt worden.

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein System, das ACEPS, vorgestellt, das zur Eindämmung der Informationsflut dient, die täglich auf einen Menschen einströmt. Das ACEPS stellt ein Email-verarbeitendes Werkzeug dar, das mit steigender Genauigkeit die Vorklassifikation ankommender Postnachrichten vornimmt. Dabei kann der Klassifikationsrahmen vom Benutzer beliebig verändert werden. Das ACEPS paßt sich durch seinen adaptiven und inkrementell lernenden Trainingsalgorithmus schnell den sich wandelnden Bedürfnissen eines Benutzers an. Seine in Kapitel 5 beschriebene Anwendung unter Einbeziehung des *Xmh*-Tools stellt zweifellos eine Erleichterung für jeden Inhaber einer *Internet*-Adresse dar. Gegen den praktischen Einsatz des Systems zum gegenwärtigen Zeitpunkt sprechen zwei Faktoren: Zum einen übersteigt der Speicherplatzbedarf des ACEPS das Ausmaß, das ein häufig benutztes Programm auf Rechnern der heutigen Generation beanspruchen kann, ohne den normalen Betrieb allzu sehr zu beeinflussen. Zum anderen verfügt das ACEPS immer noch über keine nahezu perfekte Leistung auf noch nie vorher gesehenen Emails.

Zur Steigerung der Leistungsfähigkeit des ACEPS muß daher stärker die Bedeutung des Inhalts einer Email berücksichtigt werden. Es reicht nicht aus, nur die in einer Nachricht vorkommenden Wörter zum Lernen eines Netzwerkes zu verwenden. Vielmehr muß auch die Sequenz der Wörter mit einbezogen werden. Ein Schritt in diese Richtung war bereits die Hinzunahme von Wortpaaren in die Strukturen des ACEPS-Netzwerkes. Dies erwies sich jedoch als nicht ausreichend, um neben der Stabilität auch die Generalisierungsfähigkeit des Systems zu verbessern. Weitere Versionen des ACEPS müssen verstärkt auf die Folge von Wörtern innerhalb eines Satzes bzw. innerhalb der gesamten Email eingehen.

Dennoch ermöglicht die Kombination adaptiven Lernens in Verbindung mit Backpropagation-Training hochwertige Leistungen. Ausgehend von einer Genauigkeitsrate von 47,3% ist es mit Hilfe verschiedener Netzwerkarchitekturen und Lernmethoden gelungen, die Leistungsfähigkeit des ACEPS bis auf 79,1% zu steigern. Das System ist in der Lage, Emails mit einer der menschlichen Leistung entsprechenden Genauigkeit zu klassifizieren. Darüber hinaus bietet es die gewünschte Flexibilität und ermöglicht rasche Anpassung an sich verändernde Themengebiete. Das ACEPS repräsentiert ein hybrides Email-verarbeitendes System, das folgende Merkmale liefert:

- menschenähnliche Leistung

- adaptives inkrementelles und augenblickliches *on-line* Lernen
- adaptives Wörterbuch (anstelle eines vordefinierten Vokabulars)
- keine Notwendigkeit komplizierter Grammatikregeln
- rasche Anpassung an sich ändernde Benutzerwünsche
- kein Bedarf langer Trainingsläufe auf der gesamten Datenbasis

Schließlich ist der Einsatz des ACEPS nicht nur zur Erstellung eines Systems für private Zwecke geeignet. Eine Vielzahl von Anwendungen, die über die Verarbeitung elektronischer Post hinausgehen, befassen sich mit der Einordnung konventioneller Texte in subjektiv ausgewählte semantische Kategorien. Beispiele dafür sind:

- Priorisierung und Kategorisierung von *News*
- Filtern von Informationen
- Kategorisierung von Artikeln für Bibliotheks-*Retrieval*-Systeme
- Erstellen von Textzusammenfassungen usw.

Eine der Verarbeitung von Emails ähnliche Anwendung ist die Bearbeitung von *News*. Dort kann, wie beim Einlesen neuer Mail, beim Lesen neuer Artikel einer *News*-Gruppe ein Einlesen nicht nach zeitlicher Reihenfolge, sondern nach Priorität des Benutzers erfolgen. Zusätzlich können die Artikel im Kopf mit einem Etikett versehen werden, das eine Unterklassifikation in ein Fachgebiet innerhalb der *News*-Gruppe darstellt.

Auch ist die kommerzielle Nutzung des ACEPS innerhalb eines großen Industrieunternehmens denkbar, wo an die Firma adressierte elektronische Postnachrichten vorverarbeitet werden. Die Verwendung eines Klassifizierungssystems ist dort in einem Sekretariat von Nutzen, in dem verschiedene FremdsprachensekretärInnen bzw. ÜbersetzerInnen eingehende Post in die innerhalb des Unternehmens gebräuchliche Sprache übersetzen. Niemand muß mehr ankommende Briefe vorsortieren, sondern die richtige Verteilung der Emails an die für die jeweilige Sprache zuständigen Bearbeiter wird per Rechner vorgenommen.

Weiterhin ist die Verwendung des ACEPS nicht nur auf seine Nutzung als Email-verarbeitendes Werkzeug beschränkt. Vorstellbar ist auch seine Anwendung in einer Bibliothek. Dort müssen neue Bücher oder Artikel ihrem Inhalt entsprechend in ein *Retrieval*-System eingeordnet werden, was oft sehr aufwendig ist. Das ACEPS ist nun in der Lage, mit Hilfe einer kurzen Zusammenfassung von Büchern oder Artikeln, die Kategorisierung eines Buches auszuführen. In diesem Fall ist es auch sinnvoll, eine Eingabe des ACEPS in mehr als eine Klasse einzuordnen. In das *Retrieval*-System der Bibliothek werden dann die Stichworte aller Klassen aufgenommen, deren Aktivierung über einem bestimmten Schwellwert liegt.

Eine noch utopische Anwendung ist das Erstellen von Textzusammenfassungen wie sie im vorigen Beispiel zur Eingabe des ACEPS benutzt werden. Dem ACEPS wird als Eingabe

ein Buchtext oder Artikel zur Verfügung gestellt, und das System erzeugt als Ausgabe eine Zusammenfassung dieses Textes. Dazu ist jedoch eine weitere Berücksichtigung der Semantik von Text innerhalb des ACEPS und Parsen von Sprache notwendig [Jai91, Jai92].

All diese möglichen Anwendungen des *ACEPS* lassen auf seine vielfältige Verwendbarkeit in der Zukunft schließen. Trotzdem sollten die Erwartungen an ein textverarbeitendes konnektionistisches System nicht zu hoch geschraubt werden. Es ist zu bezweifeln, ob es den Ansprüchen gerecht werden kann, von dem viele Verfechter der Künstlichen Intelligenz träumen: eine Maschine so intelligent zu machen, ihr so viel beizubringen, daß sie Menschen ersetzen kann. Dies wird sicherlich (und hoffentlich) niemals vollständig möglich sein. Maschinen werden also immer nur das ausführen, was ihnen als Verhaltensvorschrift bzw. Programm eingegeben wurde. Damit aber ist zwangsläufig vorgegeben, daß sie an gewisse Grenzen stoßen müssen. So soll diese Arbeit nun mit einem sehr treffenden Zitat von John Denker schließen

„Neural networks are the second best way of doing just about anything ...“.

Literaturverzeichnis

- [And88] J. A. Anderson und E. Rosenfeld. *Neurocomputing: Foundations of Research*. Hrsg., MIT Press, 1988.
- [Che92] Francine R. Chen und Margaret Withgott. *The Use of Emphasis to Automatically Summarize a Spoken Discourse*. Proceedings of the IEEE 1992 International Conference on Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA, Seite 229-232, März 1992.
- [Chu91] Kenneth W. Church und William A. Gale. *A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams*. Computer, Speech and Language, Vol. 5, Seite 19-54, 1991.
- [Cot87] G. W. Cottrell, P. Munro und D. Zipser. *Image Compression by Back Propagation: An Example of Extensional Programming*. ICS Report 8702, University of California at San Diego, Februar 1987.
- [Day90] Judith E. Dayhoff. *Neural Network Architectures*. Van Nostrand Reinhold, 1990.
- [Ess92] Ute Essen und Volker Steinbiss. *Cooccurrence Smoothing for Stochastic Language Modeling*. Proceedings of the IEEE 1992 International Conference on Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA. Seite 161-164, März 1992.
- [Gal92] William A. Gale und Kenneth W. Church. *Poor Estimates of Context are Worse than None*. AT&T Bell Laboratories, Murray Hill, N. J. 07974.
- [Gor90] A. L. Gorin und S. E. Levinson, A. N. Gertner, A. Ljolje und E. R. Goldman. *On Adaptive Acquisition of Language*. Proceedings of the IEEE 1990 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Albuquerque, NM, Seite 601-604, April 1990.
- [Gor91A] A. L. Gorin, S. E. Levinson und A. N. Gertner. *Adaptive Acquisition of Spoken Language*. Proceedings of the IEEE 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Seite 805-808, Mai 1991.

- [Gor91B] A. L. Gorin, S. E. Levinson, A. N. Gertner und E. R. Goldman. *Adaptive Aquisition of Language*. Computer, Speech and Language, Vol. 5, Seite 101-132, April 1991, Academic Press, Hrsg. F. Fallside, S. Levinson.
- [Gro88] Stephen Grossberg. *Neural Networks and Natural Intelligence*. 3. Auflage, MIT Press, 1988.
- [Gul88] Jürgen Gulbins. *Unix*. 3. Auflage, Springer, 1988.
- [Hau92] Roland Hausser. *Principles of Computational Morphology*. zur Veröffentlichung eingereicht, Computational Linguistics, 1992.
- [Heb49] D. O. Hebb. *The Organization of Behavior*. Wiley, 1949.
- [Hec91] Robert Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1991.
- [Her91] John Hertz, Anders Krogh und Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Lecture Notes Vol. 1, Santa Fe Institute, Studies in the Sciences of Complexity, Addison-Wesley, 1991.
- [Hop82] J. J. Hopfield. *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*. Proceedings of the National Academy of Sciences, USA Vol. 79, Seite 2554-2558, 1982.
- [Jai91] Ajay N. Jain. *Parsing Complex Sentences with Structured Connectionist Networks*. Neural Computation, Vol. 3, Seite 110-120, 1991.
- [Jai92] Ajay N. Jain, Alex Waibel und Davis S. Touretzky. *PARSEC: A Structured Connectionist Parsing System for Spoken Language*. Proceedings of the IEEE 1992 International Conference on Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA, Seite 205-208, März 1992.
- [Jel91] Frederick Jelinek, Robert L. Mercer, und Salim Roukos. *Principles of Lexical Language Modeling for Speech Recognition*. Seite 651-699, 1991.
- [Ker84] Brian W. Kernighan und Rob Pike. *The UNIX Programming Environment*. Prentice Hall, 1984.
- [Ker88] Brian W. Kernighan und Dennis M. Ritchie. *The C Programming Language*. 2. Auflage, Prentice Hall, 1988.
- [Lee88A] Kai-Fu Lee und Hsiao-Wuen Hon. *Speaker-Independent Phone Recognition Using Hidden Markov Models*. Technical Report CMU-CS-88-121, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA 15213, März 1988.
- [Lee88B] Kai-Fu Lee. *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*. PhD Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA 15213, 1988.

- [McC43] W. S. McCulloch und W. Pitts. *A Logical Calculus of Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, Vol. 5, Seite 115-133, 1943.
- [Mil91] Laura G. Miller und Allen L. Gorin. *A Structured Network Architecture for Adaptive Language Aquisition*. Proceedings of the IEEE 1992 International Conference on Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA, Seite 201-204, März 1992.
- [Min88] Marvin L. Minsky und Seymour A. Papert. *Perceptrons*. 2. Auflage, MIT Press, 1988.
- [Moo91] John E. Moody. *The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems*. in *Advances in Neural Information Processing Systems 4*, Hrsg. John E. Moody, Steven J. Hanson, Richard P. Lippman, Seite 847-854, 1991.
- [Mül91] P. H. Müller (Hrsg.). *Wahrscheinlichkeitsrechnung und mathematische Statistik*. 5. Auflage, Technische Universität Dresden, Akademie-Verlag, Berlin 1991.
- [Ney91] Herman Ney und Ute Essen. *On Smoothing Techniques for Bigram-Based Natural Language Modeling*. Proceedings of the IEEE 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Seite 825-828, Mai 1991.
- [Pie61] J. R. Pierce. *Symbols, Signals and Noise*. Harper, 1961.
- [Ros58] F. Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Psychological Review, Vol. 65, Seite 386-408, 1958.
- [Rum88] D. E. Rumelhart, G. E. Hinton, und R. J. Williams. *Learning Internal Representations by Error Propagation*. in *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*, David E. Rumelhart, James I. McClelland and the PDP Research Group. Vol. 1: Foundations. Kapitel 8, Seite 318-362, 8. Auflage, Hrsg. Jerome A. Feldman, Patrick J. Hayes, David E. Rumelhart, Computational Models of Cognition and Perception, MIT Press, 1988.
- [Schö90] E. Schöneburg, N. Hansen und A. Gawelczyk. *Neuronale Netzwerke*. Markt & Technik Verlag AG, 1990.
- [Sch85] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner und J. Makhoul. *Context-Dependent Modeling for Acoustic Phonetic Recognition of Continuous Speech*. Proceedings of the IEEE 1985 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Tampa, FL, Seite 1205-1208, April 1985.

- [Sug85] K. Sugawara, M. Nishimura, K. Toshioka, M. Okochi, T. Kaneko. *Isolated Word Recognition Using Hidden Markov Models*. Proceedings of the IEEE 1985 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Tampa, FL, Seite 1-4, April 1985.
- [Sug86] K. Sugawara, M. Nishimura, A. Kuroda. *Speaker Adaption for a Hidden Markov Model*. Proceedings of the IEEE 1986 International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 1986.
- [Tis91] Naftali Tishby und Allen Gorin. *Algebraic Learning of Statistical Associations for Language Aquisition*. AT&T Bell Laboratories, submitted for publication September 1991.
- [Was89] Philip D. Wasserman. *Neural Computing*. Von Nostrand Reinhold, 1989.
- [Wei91] Sholom M. Weiss, Casimir A. Kulikowski. *Computer Systems that learn*. Morgan Kaufmann Publishers, Inc., 1991.
- [Wid60] B. Widrow und M. E. Hoff. *Adaptive Switching Circuits*. in *1960 IRE WESCON Convention Record*. Teil 4, Seite 96-104, New York, 1960.
- [Zim90] Dieter E. Zimmer. *Die Elektrifizierung der deutschen Sprache*. Über Sprechen, Schreiben, Computer, Gehirne und Geist. Haffmans Verlag AG Zürich, 1990.

Anhang A

Beispiel-Emails

Die hier aufgeführte Email enthält in ihrer Originalform 176 Wörter, nach Bearbeitung durch die Vorverarbeitungskomponente 110 Wörter und nach Herausfiltern nicht deskriptiver Wörter 64 Wörter, was einem Vokabular von 48 voneinander verschiedenen Wörtern entspricht.

A.1 Email in Originalform

Die hier aufgeführte Email enthält in ihrer Originalform 176 Wörter.

```
Received: from Q.CS.CMU.EDU by B.GP.CS.CMU.EDU; 3 Jan 89 20:43:01 EST
Received: from CS.CMU.EDU by Q.CS.CMU.EDU; 3 Jan 89 10:53:33 EST
Received: from IDA.ORG by CS.CMU.EDU; 3 Jan 89 10:51:14 EST
Received: from csed-1 ([197.5.200.1]) by ida.org (4.0/SMI-DDN)
id AA10508; Tue, 3 Jan 89 10:48:57 EST
Received: by csed-1 (3.2/SMI-3.2)
id AA16284; Tue, 3 Jan 89 10:50:14 EST
Date: Tue, 3 Jan 89 10:50:14 EST
From: will@ida.org (Craig Will)
Message-Id: <8901031550.AA16284@csed-1>
To: connectionists@cs.cmu.edu
Subject: Copies of DARPA Request for Proposals Available
```

Copies of DARPA Request for Proposals Available

Copies of the DARPA Neural Network Request for Proposals are now available (free) upon request. This is the same text as that published December 16 in the Commerce

Business Daily, but reformatted and with bigger type for easier reading. This version was sent as a 4-page "Special supplementary issue" to subscribers of Neural Network Review in the United States.

To get a copy mailed to you, send your US postal address to either:

Michele Clouse
clouse@ida.org (milnet)

or:

Neural Network Review
P. O. Box 427
Dunn Loring, VA 22027

A.2 Email nach Bearbeitung durch die Vorverarbeitungskomponente

Die Email enthält nach Bearbeitung durch die Vorverarbeitungskomponente 110 Wörter. Nach Wegnahme der Leitweginformation reduziert sich die Wortanzahl somit auf 63% des ursprünglichen Wertes.

Subject: Copies of DARPA Request for Proposals Available

From: will@ida.org (Craig Will)

Copies of DARPA Request for Proposals Available

Copies of the DARPA Neural Network Request for Proposals are now available (free) upon request. This is the same text as that published December 16 in the Commerce Business Daily, but reformatted and with bigger type for easier reading. This version was sent as a 4-page "Special supplementary issue" to subscribers of Neural Network Review in the United States.

To get a copy mailed to you, send your US postal address to either:

Michele Clouse
clouse@ida.org (milnet)

or:

Neural Network Review
P. O. Box 427
Dunn Loring, VA 22027

A.3 Email nach Herausfiltern nicht deskriptiver Wörter

Die Email enthält nach Herausfiltern nicht deskriptiver Wörter 64 Wörter, was einem Vokabular von 48 voneinander verschiedenen Wörtern entspricht. Diese 48 Wörter stellen 36% des ursprünglichen Wortschatzes der Email in Originalform dar.

subject copies darpa request proposals available

will@ida.org craig

copies darpa request proposals available

copies darpa neural network request proposals
available free upon request

text published december commerce

business daily reformatted bigger type

reading version sent 4-page

supplementary issue subscribers neural network review

united states

get copy mailed send us postal

address

michele clouse
clouse@ida.org milnet

neural network review
box 427
dunn loring va 22027

Anhang B

Liste nicht deskriptiver Wörter (LNW)

SPECIAL WORD LIST FILE: 501 special words

! " # \$ % & ' () * + , - . / 1 10 100 11 12 13 14 15 16 17 18 19 2 20 21 22 23 24 25
26 27 28 29 3 30 31 32 33 34 35 36 37 38 39 4 40 41 42 43 44 45 46 47 48 49 5 50 51 52
53 54 55 56 57 58 59 6 60 61 62 63 64 65 66 67 68 69 7 70 71 72 73 74 75 76 77 78 79
8 80 81 82 83 84 85 86 87 88 89 9 90 91 92 93 94 95 96 97 98 99 : ; ¡ ¨ = ¿ ÿ ? @ a a
about above after again against all alone along already also although always am an and
any anybody anyone anything apart are aren't as at b back bad be because been before
behind being below beneath best better between big both but by c ca can can't cannot
certain certainly clear clearly complete could couldn't d deep did didn't different difficult
do does doesn't don't done down during e e.g each earliest early easier easiest easy eg
eight either either empty enough esp especial especially etc eventual eventually ever every
everybody everyone everything exact exactly except f far farther fast few final finally five
for foremost forth four frequent frequently from full further g good great greater greatest
h has hasn't have haven't he he'd he's her here hers herself hi himself his his how however
huge i i'd I'm if immediately in in inside into is isn't it it's its itself j just k kind l large
last late lately latest least less let little long longer longest lot lots low m may mine minus
more moreover most much must mustn't my myself n near nearly necessarily necessary
need needn't neither neither never new newer newest next nine no nobody noone nor
not now o of often old older oldest on once one only or other otherwise ought our ours
ourselves out over own p perhaps please plus precise precisely pretty probably q quite
r rare rarely rather recent recently s same separate seven several she she'd she's short
shorter shortest should shouldn't simple simpler simplest since six slow slowly small so
soas some somebody someone something sometime sometimes soon sort sought special
such sure surely t ten thank thanks that the their theirs themselves then there there's
therefore these they they'd they're this those though three through throughtout till time
tiny to together total toward towards twice two typical typically u under unfortunately
unless until up v very vs w was wasn't we we'd we're well were weren't what whatever
when when whenever where whereas wherever whether which while whoever why wide
will with within without won't worse worst would wouldn't x y yes yet you you'd you're
your your yours yourself yourselves z [\] - ' { — }

Anhang C

Wortlisten

Im folgenden werden Listen der am höchsten gewichteten Wörter pro Themengebiet vorgestellt.

C.1 Konferenzankündigungen

workshops

held

darpa

committee

co-chairs

camera-ready

non-members

congress

workshop

symposium

participation

registration

annual

programme

deadline

sessions

chairman

pre-registration
conference
icnn'91
tutorial
on-campus
ijcnn'91
call
nips*92
tutorial/workshop
papers
submissions

C.2 Fachliche Diskussion

tractability
separability
question
connectionist
domain
levels
result
think

C.3 Stellenangebote

openings
m.a
positions
assistant
post-doctoral
state-funded

associate
programmer
permanent
position
postdoctoral
tenure-track
offers
hiring
fellowships
job
postdoc
tenure
salary
qualified
professors
postdocs
experience
applicants
post-doc
opportunity/affirmative
employer
non-canadian
nominees/applicants

C.4 Unwichtiges

unsubscribe
subscribe
offensive
limericks
missionaries

subscription

stop

signoff

sexist

moved

joke

C.5 Veröffentlichungen

remote-file

local-file

pub/neuroprose

editor-in-charge

hardcopies

newsgroups

abstract

report

technical

notes

forthcoming

tech

reprints

postscript

publishes

printing

journal

available

volume

submitted

Anhang D

Beispiele falsch klassifizierter Emails

D.1 In „Konferenzankündigungen“ eingeordnete Emails der Klasse „Unwichtiges“

D.1.1 Beispiel 1

```
Received: from q.cs.cmu.edu by B.GP.CS.CMU.EDU id aa07478; 9 Apr 90 17:08:15 EDT
Received: from Q.CS.CMU.EDU by Q.CS.CMU.EDU id aa04684; 9 Apr 90 15:07:40 EDT
Received: from cs.cmu.edu by Q.CS.CMU.EDU id aa04681; 9 Apr 90 14:48:51 EDT
Received: from [128.6.37.16] by CS.CMU.EDU id aa12205; 9 Apr 90 14:21:31 EDT
Received: by caip.rutgers.edu (5.59/SMI4.0/RU1.1/3.04)
id AA17500; Mon, 9 Apr 90 14:20:43 EDT
Date: Mon, 9 Apr 90 14:20:43 EDT
From: ananth sankar <sankar@caip.rutgers.edu>
Message-Id: <9004091820.AA17500@caip.rutgers.edu>
To: connectionists@CS.CMU.EDU
```

D.1.2 Beispiel 2

```
Received: from q.cs.cmu.edu by B.GP.CS.CMU.EDU id aa21970;
19 Nov 90 21:52:44 EST
Received: from Q.CS.CMU.EDU by Q.CS.CMU.EDU id aa02884; 19 Nov 90 9:23:21 EST
Received: from b.gp.cs.cmu.edu by Q.CS.CMU.EDU id aa02882; 19 Nov 90 9:13:47 EST
Received: from b.gp.cs.cmu.edu by B.GP.CS.CMU.EDU id aa10824;
19 Nov 90 9:12:29 EST
From: Connectionists-Request@CS.CMU.EDU
To: connectionists@CS.CMU.EDU
reply-to: David Haussler <haussler@SATURN.UCSC.EDU>
Subject: call for papers for COLT '91
Date: Mon, 19 Nov 90 09:12:22 EST
```

Message-ID: <10821.659023942@B.GP.CS.CMU.EDU>

Sender: Connectionist.Research.Group@B.GP.CS.CMU.EDU

Return-Path: <@CS.CMU.EDU:haussler@saturn.ucsc.edu>

D.1.3 Beispiel 3

Received: from Q.CS.CMU.EDU by B.GP.CS.CMU.EDU; 27 Feb 89 16:35:52 EST
Received: from ri.cmu.edu by Q.CS.CMU.EDU id aa07989; 27 Feb 89 11:01:40 EST
Received: from CENTRAL.CIS.UPENN.EDU by RI.CMU.EDU; 27 Feb 89 10:59:14 EST
Received: from CIS.UPENN.EDU by central.cis.upenn.edu
id AA07563; Mon, 27 Feb 89 08:50:12 -0500
Posted-Date: Sun, 26 Feb 89 18:45 EST
Message-Id: <8902271350.AA07563@central.cis.upenn.edu>
Date: Sun, 26 Feb 89 18:45 EST
From: Lokendra Shastri <Shastri@cis.upenn.edu>
Subject: Rules and variables in a connectionist system
To: connectionists@RI.CMU.EDU

D.1.4 Beispiel 4

Received: from q.cs.cmu.edu by B.GP.CS.CMU.EDU id ab13903; 1 Aug 89 14:56:01 ED
Received: from garnet.berkeley.edu by Q.CS.CMU.EDU id aa05526;
1 Aug 89 13:41:29 EDT
Received: by garnet.berkeley.edu (5.57/1.32)
id AA11138; Tue, 1 Aug 89 10:41:15 PDT
Date: Tue, 1 Aug 89 10:41:15 PDT
From: movellan@garnet.berkeley.edu
Message-Id: <8908011741.AA11138@garnet.berkeley.edu>
To: connectionists@Q.CS.CMU.EDU
Subject: International Joint Conference on Neural Networks

Does anybody know the address for paper submissions to the next
IJCNN ?

D.2 Fehlleistung der Vorklassifikation

Received: from q.cs.cmu.edu by B.GP.CS.CMU.EDU id ah27090; 29 Jan 91 0:54:32 EST
 Received: from Q.CS.CMU.EDU by Q.CS.CMU.EDU id aa02841; 28 Jan 91 14:01:33 EST
 Received: from cs.cmu.edu by Q.CS.CMU.EDU id aa02835; 28 Jan 91 13:50:08 EST
 Received: from grieg.cs.colostate.edu by CS.CMU.EDU id aa22242;
 28 Jan 91 13:46:36 EST
 Received: by grieg.CS.ColoState.Edu (4.0/SMI-DDN)
 id AA04012; Mon, 28 Jan 91 11:46:28 MST
 Date: Mon, 28 Jan 91 11:46:28 MST
 From: charles anderson <andercha@grieg.CS.ColoState.EDU>
 Message-Id: <9101281846.AA04012@grieg.CS.ColoState.Edu>
 To: connectionists@CS.CMU.EDU
 Subject: Call for Papers for 1991 Machine Learning Workshop

CALL FOR PAPERS
 1991 MACHINE LEARNING WORKSHOP
 Northwestern University June 27-29, 1991

CONSTRUCTIVE INDUCTION

Selection of an appropriate representation is critical to the success of most learning systems. In difficult learning problems (e.g., protein folding, word pronunciation, relation learning), considerable human effort is often required to identify the basic terms of the representation language. Constructive induction offers a partial solution to this problem by automatically introducing new terms into the representation as needed. Automatically constructing new terms is difficult because the environment or teacher usually provides only indirect feedback, thus raising the issue of credit assignment. However, as learning systems face tasks of greater autonomy and complexity, effective methods for constructive induction are becoming increasingly important.

The objective of this workshop is to provide a forum for the interchange of ideas among researchers actively working on constructive induction issues. It is intended to identify commonalities and differences among various existing and emerging approaches such as knowledge-based term construction, relation learning, theory revision in analytic systems, learning of hidden-units in multi-layer neural networks, rule-creation in classifier systems, inverse resolution, and qualitative-law discovery.

Submissions are encouraged in the following topic areas:

- o Empirical approaches and the use of inductive biases

- o Use of domain knowledge in the construction and evaluation of new terms
- o Construction of or from relational predicates
- o Theory revision in analytic-learning systems
- o Unsupervised learning and credit assignment in constructive induction
- o Interpreting hidden units as constructed features
- o Constructive induction in human learning
- o Techniques for handling noise and uncertainty
- o Experimental studies of constructive induction systems
- o Theoretical proofs, frameworks, and comparative analyses
- o Comparison of techniques from empirical learning, analytical learning, classifier systems, and neural networks

Send six copies of paper submissions (3000 word maximum) to Christopher Matheus, GTE Laboratories, 40 Sylvan Road, MS-45, Waltham MA 02254 (matheus@gte.com). Submissions must be received by February 1, 1991. Include a cover page with authors' names, addresses, phone numbers, electronic mail addresses, paper title, and a 300 (maximum) word abstract. Do not indicate or allude to authorship anywhere within the paper. Acceptance notification will be mailed by March 31, 1991. Accepted papers will be allotted four two-column pages for publication in the Proceedings of the 1991 Machine Learning Workshop.

Organizing Committee:

Christopher Matheus, GTE Laboratories
George Drastal, Siemens Corp. Research
Larry Rendell, University of Illinois

Program Committee:

Chuck Anderson, Colorado State
Gunar Liepins, Oak Ridge National Lab.
Douglas Medin, University of Michigan
Paul Utgoff, University of Massachusetts