Erkennung und Transkription Neuer Wörter in der Spracherkennung

Diplomarbeit von

Bernhard Suhm
(bsuhm@cs.cmu.edu)

am
Institut für Logik, Komplexität und Deduktionssysteme
Universität Karlsruhe

28. April 1993

Hauptreferent: Prof. Alex H. Waibel
Betreuerin: Dipl. Phys. Monika Woszczyna

Inhaltsverzeichnis

Hiermit erkläre ich, daß ich die vorliegende Diplomarbeit selbständig und ohne unzulässige Hilfsmittel angefertigt habe. Alle verwendeten Quellen sind im Literaturverzeichnis aufgeführt.

Bernhard Suhm
Pittsburgh, 28. April 1993

Zusammenfassung

Forschungs–Spracherkennungssysteme sind daraufhin ausgelegt, Wörter innerhalb eines bestimmten Vokabulars erkennen zu können. Wenn daher in Spracheingaben Wörter außerhalb dieses Vokabulars vorkommen, führt dies im allgemeinen zur Ausgabe entstellter Satzhypothesen. Ein Spracherkenner mit der Fähigkeit, *neue Wörter* zu detektieren und gegebenenfalls phonetisch zu transkribieren, könnte dagegen auch solche Spracheingaben verarbeiten.

Die Untersuchung des *Neue–Wort Problems* und die Implementierung eines *Neuen–Wort Modells* im Erkennungsmodul des JANUS–Systems waren Hauptziele dieser Arbeit.

Zunächst wurden einige Eigenschaften *neuer Wörter* im Hinblick auf die Entwicklung eines *Neuen–Wort Modells* anhand großer Texte untersucht. Anschließend wurden verschiedene Modellierungs–Möglichkeiten implementiert und auf Sprachdaten des englischen *Conference Registration Task* getestet.

Wesentliche Merkmale des vorgestellten Ansatzes sind die Erkennung sowohl bekannter als auch neuer Wörter in einem Such–Durchlauf und der Einsatz eines Sprachmodells für neue Wörter.

Abstract

State–of–the–art systems for continuous speech recognition are designed to recognize words within a certain vocabulary. If speech utterances contain out–of–vocabulary words the system will try to map other words from the vocabulary onto all parts of the utterance, leading to major recognition errors. However, if a speech–recognition system could be equipped with the ability to detect *new–words* it could handle their occurance gracefully.

Primary goal of this thesis was to investigate the nature of the *new–word problem* and to implement a *new–word model* using the recognition engine of the JANUS–system.

Some features of *new–words* were examined using large texts, trying to develop a suitable approach to model *new–words*. Different *new–word models* were implemented and compared using speech–recordings from the English *Conference Registration* task.

The main features of this approach are the recognition both of known and new words in one search, and the use of a language model to detect *new–words*.

Danksagung

An erster Stelle möchte ich Prof. Waibel danken, der mir den Aufenthalt an der *Carnegie Mellon University* zur Anfertigung der Diplomarbeit ermöglicht hat. Diese außergewöhnliche Arbeitsumgebung hat zusätzliche Möglichkeiten eröffnet. Als betreuender Professor hat er sich immer wieder Zeit für Diskussionen und Anregungen genommen.

Besonderer Dank gilt auch meiner Betreuerin Monika Woszczyna für ihre wertvolle Hilfestellung, von der „Eingewöhnung" in das JANUS–System bis zum Korrekturlesen dieser Ausarbeitung. Größtenteils mußte dies in Form einer „Fernbetreuung" über E–Mail abgewickelt werden.

Viele andere haben auf unterschiedliche Art und Weise zum Gelingen dieser Arbeit beigetragen: Deb Roy (Software für statistische Grammatiken), Wayne Ward und Bob Weide (Zugang zu Datenbasen), Arthur McNair (Systembetreuung), Tilo Sloboda und Jürgen Dingel (Korrekturlesen), Cindy Wood (Sprachaufnahmen). Bedanken möchte ich mich auch bei Radha Rao und Barbara Moore, die mir beim Nehmen der bürokratischen und unbürokratischen Hürden des Alltags am *Center for Machine Translation* zur Seite standen.

Mehr als Dank verbindet mich mit meinen Eltern, ohne deren finanzielle Unterstützung der Aufenthalt an der *Carnegie Mellon University* nicht möglich gewesen wäre.

Kapitel 1

Einführung

Im Gebiet der maschinellen Spracherkennung versucht man, auf Anwendungen aus dem Alltag überzugehen, Anstelle abgelesener Sätze aus eng umgrenzten Anwendungssituationen soll in Zukunft **spontane Sprache**, darunter versteht man natürlich–sprachliche Äußerungen, maschinell erkannt werden.

Im folgenden Abschnitt werden anhand von Beispielanwendungen die Besonderheiten spontaner Sprache erläutert. Anschließend werden auf die Möglichkeiten und Grenzen von Spracherkennungssystemen eingegangen. Abschließend wird das Problem der Erkennung neuer Wörter als eines der Probleme im Zusammenhang mit der Erkennung spontaner Sprache vorgestellt.

1.1 Herausforderung spontaner Sprache

Anhand von zwei Anwendungen zukünftiger Spracherkennungssysteme sollen die Besonderheiten spontaner Sprache aufgezeigt werden, wie sie beim Einsatz von Sprache als Kommunikationsmedium an der Mensch–Maschine Schnittstelle auftreten.

In einem von der Firma Apple[1] für Werbezwecke erstellten Video wird der sogenannte *Knowledge Navigator* vorgestellt. Dieses Video kann man als optimistischen Ausblick in die Zukunft von sprachverarbeitenden Systemen ansehen. Der *Knowledge Navigator* ist in der Lage, dem Benutzer eine Vielfalt von Diensten anzubieten. Diese reichen von Recherchen, über Terminplanung, bis hin zur Übersetzung von Gesprächen. Die gesamte Interaktion zwischen dem System und dem Benutzer wird über einen natürlich–sprachlichen Dialog abgewickelt. Anfragen an das System können in Sätzen formuliert werden, wie man sie auch einem Arbeitskollegen stellen würde. Der *Knowledge Navigator* versucht, Unklarheiten und Mißverständnisse durch „Nachfragen" aufzuklären. Zur Ausführung des gewünschten Dienstes zusätzlich benötigte Informationen werden vom Benutzer in Form von gezielten Fragen angefordert.

[1] Apple Computer Inc, Cupertino, California (USA)

Eine andere weniger futuristisch anmutende Anwendung könnte ein System sein, das mehrsprachige Telefongespräche oder Video–Konferenzschaltungen unterstützt. Jeder Gesprächs–Partner spricht und hört in seiner jeweiligen Muttersprache. Man muß also weder zum Verständnis des anderen noch zum Ausdrücken der eigenen Gedanken in eine andere Sprache umdenken. In einem solchen System sind neben der Aufgabe der Spracherkennung zusätzlich maschinelle Übersetzung und Sprachsynthese zu bewältigen.

Eine maschinell unterstützte mehrsprachige Konferenzschaltung wurde im Rahmen des JANUS–Projektes[2] für die eng eingegrenzte Aufgabenstellung der *Conference Registration Task*[3] bereits getestet.

Solche und ähnliche Anwendungen beinhalten Erkennung spontaner Sprache. Die folgenden Problemkreise zeigen, daß diese Aufgabe eine Herausforderung ist. Sie wurden aus einem Vergleich abgelesener mit spontaner Sprache anhand eines simulierten Mensch/Maschine Dialogs in [4] entnommen.

- Akustik:
  Sprachliche Äußerungen können durch Husten oder Räuspern unterbrochen sein. Mitten im Satz treten Sprechpausen auf, die mit Lauten wie „Hmm" gefüllt sein können. Zusätzlich können Hintergrundgeräusche zu einem Problem werden, insbesondere wenn man in Anwendungen wie dem *Knowledge Navigator* dem Benutzer Bewegungsfreiheit einräumen will und Raummikrophone einsetzt.

- Grammatik:
  Äußerungen in einem Telefongespräch oder freiformulierte Anfragen an den *Knowlegde Navigator* sind nicht immer grammatikalisch wohlgeformte Sätze, sondern es treten grammatikalische Fehler, Wiederholungen oder Neuformulierungen „verunglückter" Teilsätze auf.

- Vokabular:
  Selbst bei Einschränkung auf bestimmte Themenbereiche ist es kaum möglich, ein Vokabular festzulegen, das alle in spontanen Spracheingaben auftretenden Wörter abdeckt. Als Themenbereiche kann man zum Beispiel Konferenzanmeldungen, Hotelreservierungen, Katalogbestellungen oder Reisebuchungen nennen.

1.2 Möglichkeiten und Grenzen von Spracherkennungssystemen

Die Forschung auf dem Gebiet der maschinellen Spracherkennung hat in den vergangenen zwei Jahrzehnten bemerkenswerte Fortschritte erzielt. Anwendungssysteme wie ein sprachgesteuertes Operationsmikroskop der Firma Carl Zeiss gibt es dagegen nur ganz wenige. Dies liegt u.a. an dem erheblichen Aufwand an Hardware und Software für Spracherkennungssysteme. Im folgenden wird auf die Möglichkeiten und Grenzen dieser Systeme kurz eingegangen.

[2] JANUS ist eine Kooperation der Carnegie–Mellon University (Pittsburgh) und Universität Karlsruhe.
[3] Eine Beschreibung dieser und anderer Anwendungen befindet sich in Anhang A.

1.2.1 Kontinuierliche Sprache

Die ersten Spracherkennungssysteme waren Einzelworterkenner. Diese konnten einzeln gesprochene und durch Pausen getrennte Wörter erkennen. Inzwischen gilt Einzelworterkennung aber als abgeschlossenes Forschungsthema. Es gibt Erkennungssysteme für kontinuierliche Sprache von verschiedenen Forschungsgruppen.

Die Erkennung kontinuierlicher Sprache ist erheblich schwieriger als Einzelworterkennung. Dies liegt vor allem an den folgenden Eigenschaften kontinuierlicher Sprache:

- Zwischen einzelnen Wörtern werden in der Regel keine Sprechpausen gemacht.
- Durch Koartikulation wird die Aussprache eines Lauts durch die umgebenden Laute mehr oder weniger stark verändert.
- Unterschiede in der Aussprache von Wörtern durch Betonung und Prosodie.

Der letzte Punkt ist für die englische Sprache von besonderer Bedeutung. Die inhaltstragenden Wörter werden sehr stark betont. Funktionsworte, wie Artikel, Präpositionen, Hilfsverben, Negationen, werden dagegen meist unbetont und unter Auslassung ganzer Silben ausgesprochen.

1.2.2 Sprecher(un)abhängigkeit

Ein sprecherabhängiges System ist auf bestmögliche Erkennungsleistung für einen bestimmten Sprecher hin optimiert. Dies kann durch die Einstellung der Systemparameter erfolgen. Sprecherunabhängige Systeme sind dagegen auf eine gute durchschnittliche Erkennungsleistung bei beliebigen Sprechern ausgelegt. Sprecherunabhängigkeit ist eine Herausforderung aufgrund der großen Variabilität der Sprache zwischen verschiedenen Sprechern. Spracherkenner sind heutzutage meist als sprecherunabhängige Systeme ausgelegt.

Eine weitere Möglichkeit ist, die Parameter des Systems in einer Lernphase auf einen neuen Sprecher hin zu optimieren. In diesem Fall spricht man von Sprecheradaption.

1.2.3 Sprachmodell

Sprachmodelle in der Form von Grammatiken werden dazu benutzt, die zugelassenen Wortfolgen festzulegen. Die Schwierigkeit einer Erkennungsaufgabe hängt stark davon ab, welche Zwangsbedingungen an zulässige Satzhypothesen durch die Grammatik gestellt werden.

1.2.4 Vokabulargröße

Die verschiedenen Anwendungen unterscheiden sich neben dem Sprachmodell auch in der Größe des zugrundeliegenden Vokabulars. Kleine Vokabulare bewegen sich in der Größenordnung bis zu einigen hundert Wörtern, große Vokabulare umfassen dagegen einigen tausend bis zu mehreren zehntausend Wörtern. Diese Größenordnungen haben sich in den

letzten Jahren aber sehr zu größeren Vokabularen hin verschoben; so galten vor wenigen Jahren einige hundert Wörter noch als große Vokabulare.

Mit zunehmender Vokabulargröße erschweren einige zusätzliche Probleme die Erkennungsaufgabe. Hier sind insbesondere die zunehmende Verwechselbarkeit der Wörter innerhalb des Vokabulars und die Komplexität der Suche nach der Satzhypothese zu nennen.

1.2.5 Erkennungsleistung einiger Beispielsysteme

Anhand von drei Spracherkennungssystemen verschiedener Forschungsgruppen sollen die in diesem Abschnitt gemachten Aussagen durch einige Testergebnisse illustriert werden. Alle hier aufgeführten Systeme erkennen kontinuierliche Sprache. Eine Beschreibung der Anwendungen *Conference Registration* und *Resource Management* befindet sich im Anhang A.

JANUS ist ein sprecherunabhängiges System, das kontinuierlich gesprochenes Englisch oder Deutsch übersetzt in gesprochenes Deutsch, Englisch oder Japanisch. Das Spracherkennungsmodul von JANUS erzielt auf dem englischen *Conference Registration Task* mit einem Vokabular von 408 Wörtern und einer Grammatik mit Perplexität[4] 18 eine Worterkennungsrate von 91,5%, wie in [5] veröffentlicht. Auf dem *Resource Management Task* mit einem Vokabular von 997 Wörtern bei einer Perplexität von 60 erreicht JANUS eine Worterkennunsrate von 91,3%.

An der Carnegie Mellon University wurde das SPHINX Spracherkennungssystem [2] entwickelt. Als sprecherunabhängiges System erreicht es auf dem *Resource Management Task* bei der Perplexität 60 eine Worterkennungsraten 94,7%. Bei Verzicht auf eine Grammatik (bei Perplexität 997) sinkt die Worterkennungsrate auf 73,6%; bei Verwendung einer Grammatik mit Perplexität 20 steigt sie auf 96,2%. Dies zeigt die Bedeutung der Grammatik für eine gute Erkennungsleistung.

Das BYBLOS System von BBN [6] erreicht in seiner sprecherabhängigen Version auf dem *Resource Management Task* eine Worterkennungsrate von 92,5% bei einer Perplexität von 60.

1.3 Das Problem neuer Wörter

Erkennungssysteme für kontinuierlich gesprochene Sprache versuchen, Wörter aus einem a priori festzulegenden Vokabular auf alle Teile der Spracheingabe abzubilden. Wenn daher in der Spracheingabe **neue Wörter**—dies sind Wörter außerhalb des Vokabulars des Systems—vorkommen, erscheinen an ihrer Stelle Folgen von irgendwelchen anderen Wörtern, die im Vokabular enthalten sind. Dies führt also zu Erkennungsfehlern.

[4] Perplexität ist ein informationstheoretisches Maß für die Schwierigkeit einer Erkennugsaufgabe (siehe Abschnitt 2.3.2)

Folgendes Beispiel anhand des JANUS–Erkennungssystems möge dies verdeutlichen: Die Spracheingabe

„My Name is Christopher Ohara. My wife will be coming too."

wobei die Wörter „Christopher" und „Ohara" neue Wörter sind, führt zur Ausgabe der Satzhypothese:

MY NAME IS THERE IS THERE WHEN WILL BE COMING TOO

An diesem Beispiel sind folgende Punkte typisch für das **Neue–Wort Problem**:

- Bis zum Auftreten des ersten neuen Wortes hat der Erkenner die richtigen Wörter gefunden.
- Die neuen Wörter wurden durch bekannte Wörter ersetzt. Eine akustische Ähnlichkeit zwischen der tatsächlichen Eingabe und der gefundenen Hypothese braucht, je nach Größe des Vokabulars und Perplexität der Grammatik, nicht mehr zu bestehen.
- Nach einigen Erkennungsfehlern kann der Erkenner sich wieder erholen und erkennt wieder die richtigen Wörter.

Dieses Verhalten des Erkennungssystems ist unerwünscht: In der Anwendungssituation der Kommunikation eines menschlichen Benutzers mit einem Computer über ein Spracherkennungssystem kann es dazu führen, daß der Mensch annimmt, das System habe ihn mißverstanden. Er wird dann versuchen, dem System durch Wiederholen desselben Satzes weiterzuhelfen, was offensichtlich in einer Sackgasse endet. In einem System wie JANUS, bei dem sich an die Spracherkennung eine maschinelle Übersetzung anschließt, können durch neue Wörter entstellte Satzhypothesen nicht mehr übersetzt werden.

Es stellt sich die Frage, wie neue Wörter in einem Spracherkennungssystem behandelt werden sollen. Ein erster Schritt ist, Auftrittsorte neuer Wörter innerhalb der Spracheingabe zu detektieren. Daran schließt sich die Möglichkeit an, für das neue Wort eine phonetische Transkription auszugeben oder es akustisch wiederzugeben.

In Falle eines Sprach–zu–Sprach Übersetzungssystems könnte das Übersetzungsmodul befähigt werden, korrekt erkannte Teilsätze zu übersetzen. Mittels einer Interaktion könnte der Benutzer ferner dazu aufgefordert werden, das neue Wort über die Tastatur einzugeben. Häufig auftretende neue Wörter könnten im Rahmen einer adaptiven Vokabularanpassung in das Vokabular des Erkenners aufgenommen werden. Wenn ein spezielles Buchstabier–Erkennungssystem zur Verfügung steht, ist als weitere Alternative denkbar, den Benutzer zum Buchstabieren des neuen Wortes aufzufordern.

Es ist also wünschenswert, über ein Spracherkennungssystem zu verfügen, das in der Spracheingabe vorhandene neue Wörter als solche behandeln kann.

Kapitel 2

Grundlagen

In diesem Kapitel werden diejenigen Begriffe und Algorithmen aus dem Bereich der maschinellen Spracherkennung vorgestellt, die für die Entwicklung eines Verfahrens zur Erkennung neuer Wörter von Bedeutung sind.

Im ersten Abschnitt wird beschrieben, wie man vom akustischen Signal zu einer Darstellung auf Phonemebene gelangt. Die phonetische Modellierung hat entscheidenden Einfluß auf die Erkennungsleistung des Neuen–Wort Erkenners. Anschließend wird im zweiten Abschnitt erläutert, wie man ausgehend von den Phonembewertungen nach derjenigen Folge von Wörtern sucht, die am „besten" mit der tatsächlichen Spracheingabe übereinstimmt. Die Implementierung eines Neuen–Wort Erkenners setzt in dieser Suche an. Im darauffolgenden Abschnitt werden statistische Grammatiken vorgestellt. Sie werden in Spracherkennungssystemen als Sprachmodell (*language model*) zur Einschränkung der Syntax der zu erkennenden Sätze benutzt. Im Neuen–Wort Erkenner werden statistische Grammatiken sowohl auf der Phonem– als auch auf der Wortebene eingesetzt.

2.1 Phonembewertungen

Algorithmen zur Analyse der Sprachsignale bestimmen aus dem Sprachsignal, das durch Spektralkoeffizienten repräsentiert wird, Bewertungen für jedes Phonem oder Phonemsegment. Diese Bewertungen werden in der Suche nach der Satzhypothese dazu benutzt, die DP–Matrix zu initialisieren (mehr hierzu in Abschnitt 2.2).

Hier wird ein auf *Learning Vector Quantization (LVQ)* [7] beruhendes Verfahren beschrieben. Der im Rahmen dieser Diplomarbeit benutzte Erkenner verwendet solche LVQ–Phonem–Modelle. Für andere Ansätze wie *Linked Predictive Nerual Networks (LPNN)* und *Multi–Stage Time Delayed Neural Networks (MS–TDNN)* wird auf die Literatur [8, 9] verwiesen.

2.1.1 Kontextunabhängiges LVQ–Phonem–Modell

Das in [10] vorgestellte LVQ–Phonem–Modell kann als ein hybrider neuronaler und statistischer Ansatz charakterisiert werden. Jedes Phonem wird unabhängig von dem umge-

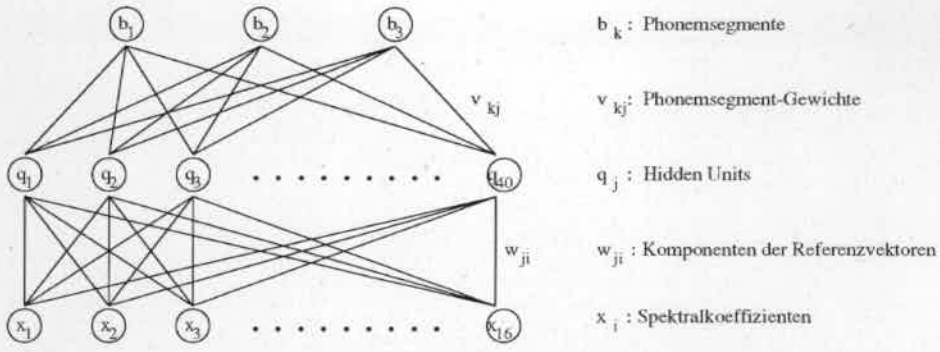

Abbildung 2.1: Architektur des LVQ–Netzes

benden Phonem–Kontext modelliert.

Merkmalsvektoren des Sprachsignals werden in sogenannten LVQ–Netzen für jedes Phonem mit prototypischen Merkmalsvektoren verglichen. Für jedes Phonem gibt es zwei voneinander unabhängige LVQ–Netze: Das erste Netz erhält die Spektralkoeffizienten des Sprachsignals zu einem Zeitpunkt. Das zweite Netz erhält die Differenz der Spektralkoeffizienten des vorangegangen und des nachfolgenden Zeitpunktes. Dadurch kann auch die Dynamik der Spracheingabe modelliert werden.

Temporale Abhängigkeiten werden in ein Phonem–Modell abgebildet, das auch als *Hidden Markov Modell (HMM)* gedeutet werden kann.

Die Architektur eines LVQ–Netzes ist in Abbildung 2.1 dargestellt. Die Gewichte w_{ji} zu den Hidden–Units repräsentieren die Referenzvektoren, die während der Trainingsphase als prototypische Merkmalsvektoren gelernt werden.[1] Die Knoten der Ausgabeschicht entsprechen den möglichen Phonemsegmenten Anfang, Mitte und Ende. Ihre Aktivierung wird als Bewertung für das entsprechende Phonemsegment gedeutet. Die Gewichte v_{kj} kann man als Wahrscheinlichkeit interpretieren, daß im Phonemsegment k der Referenzvektor j beobachtet werden kann.

In der Trainingsphase werden die Gewichte w_{ji} durch den LVQ2–Algorithmus auf eine minimale Anzahl von Fehlklassifikationen hin optimiert. Hierbei wird eine Veränderung der Gewichte nur vorgenommen, wenn der Trainingsvektor falsch klassifiziert wurde. Die Gewichte werden dann derart verändert, dass der Referenzvektor der korrekten Klasse auf den Trainingsvektor hin bewegt wird, wohingegen der Referenzvektor der fälschlicherweise gewählten Klasse vom Trainingsvektor weg bewegt wird. Die Gewichte v_{kj} werden dagegen durch die relative Häufigkeit geschätzt, daß im Training Phonemsegment k im Zusammenhang mit Referenzvektor j beobachtet wurde.

Während der Suche muß für jeden Zeitpunkt der Spracheingabe eine Bewertung für jedes Phonem bestimmt werden. Hierzu werden die Bewertungen des jeweiligen ersten und zweiten LVQ–Netzes gewichtet addiert.

In Abbildung 2.2 ist ein Phonem–Modell mit 6 Zuständen zusammen mit der Ankopplung an das zugehörige LVQ–Netz dargestellt. Man beachte, daß jedem Phonemsegment zwei

[1] Die Menge aller Referenzvektoren bilden ein sogenanntes Vektor–Kodebuch

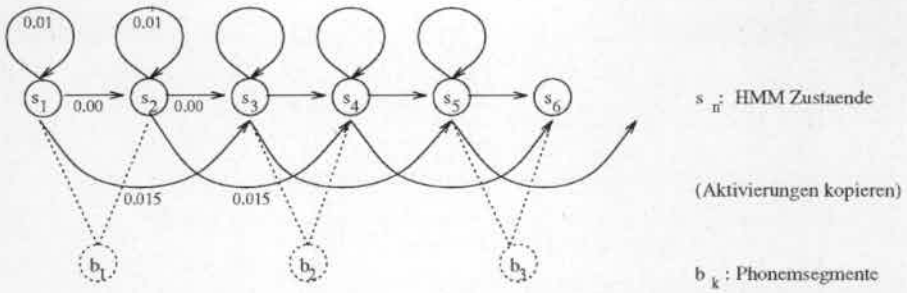

Abbildung 2.2: Phonem–Modell mit 6 Zuständen und Ankopplung an das zugehörige LVQ–Netz

benachbarte Zustände zugeordnet sind. Die zugehörigen Aktivierungen werden in die entsprechenden Zustände kopiert. Die angedeuteten Strafen für Transitionen zwischen Zuständen werden erst in der Suche nach der Satzhypothese verwendet.

2.1.2 Kontextabhängige Phonem–Modelle

In der Einleitung 1.2.1 wurde bereits Koartikulation zwischen Wörtern erwähnt. In ähnlicher Weise wird auch die Aussprache einzelner Phoneme von den umgebenden Phonemen beeinflußt.

Das im letzten Abschnitt beschriebene LVQ–Phonem–Modell ignoriert die verschiedenen Kontexte der Phoneme. In diesem Fall spricht man von **Monophonen**. Die Koartikulation zwischen Phonemen kann modelliert werden, indem man jedes Phonem mit dem linken und rechten Nachbarphonem zu einer Einheit zusammenfasst. Eine solche Einheit bezeichnet man als **Triphon**.

Aufgrund der großen Zahl verschiedener Triphone[2] stellt sich dann jedoch das Problem ausreichender Trainingsdaten für eine große Zahl von Parametern.[3] Viele der möglichen Triphone kommen in den Trainingsdaten selten oder überhaupt nicht vor.

Eine Reduktion der Anzahl von Triphonen und damit der im Training zu lernenden Parameter kann man erreichen, indem ähnliche Phonemkontexte zu *generalisierten* Triphonen zusammengefasst werden. Das Auffinden der ähnlichen Phonemkontexte kann automatisch durch einen Ballungs–Algorithmus erfolgen, wie in [2] beschrieben.

Die Architektur der LVQ–Netze für kontextabhängige Triphone ist im Prinzip dieselbe wie für die im letzten Abschnitt vorgestellten kontextunabhängigen Monophone. Für jedes Phonem wird eine Menge kontextunabhängiger, prototypischer Referenzvektoren trainiert. Die Phonemsegment–Gewichte werden jedoch (kontextabhängig) durch die relativen Häufigkeiten geschätzt, daß Triphon–Segmente im Zusammenhang mit diesen kontextunabhängigen Referenzvektoren im Training auftreten.

---

[2] Aus N verschiedenen Phonemen kann man N^3 Triphone bilden, wobei für das englische JANUS–System $N = 40$.

[3] Für die im letzten Abschnitt beschriebenen LVQ–Netze sind dies 760 Parameter je Netz.

2.2 Suche der Satzhypothesen

2.2.1 Von Phonembewertungen zu Satzhypothesen

Ein Algorithmus zur Suche von Satzhypothesen muß folgende Grundprobleme lösen:

- Erkennung der gesprochenen Wörter:
  Das Erkennungsproblem wird in bisherigen Spracherkennungssystemen eingeschränkt auf diejenigen Wörter, die a priori anhand des Wörterbuchs bekannt sind.

- Auffinden der Wortgrenzen:
  Da in kontinuierlich gesprochener Sprache in der Regel zwischen den Wörtern keine Sprechpausen gemacht werden, muß der Suchalgorithmus auch bestimmen, an welchen Zeitpunkten ein Wort aufhört und das folgende Wort beginnt.

- Nicht–lineare zeitliche Längenanpassung:
  Aufgrund der hohen Variabilität gesprochener Sprache stellt sich beim Vergleich der Referenzmuster mit dem Sprachsignal überdies das Problem einer nicht–linearen zeitlichen Längenanpassung.[4]

Diese Suche wird aus Effizienzgründen durch Zwangsbedingungen stark eingeschränkt:

- auf der Wortebene durch Einschränkung der zulässigen Wörter auf ein Vokabular

- auf der Satzebene durch ein Sprachmodell.

2.2.2 *Dynamic Time Warping* für kontinuierlich gesprochene Sprache

Im folgenden wird der von Ney in [11] vorgestellte *one-stage* Algorithmus beschrieben. Er kann aus dem Prinzip des Dynamischen Programmierens hergeleitet werden. Die Satzhypothese wird in einem Durchlauf durch das Sprachsignal gefunden.

Der Suchraum wird auf die sogenannte **DP–Matrix** abgebildet. An der x–Achse sind die Merkmalsvektoren des Sprachsignals für jeden elementaren Zeitabschnitt aufgetragen. An der y–Achse muß man sich die Wortmodelle der Wörter innerhalb des Vokabulars vorstellen. Die Matrix selbst wird mit Phonem–Bewertungen initialisiert, die mit Hilfe der im Abschnitt 2.1 beschriebenen Verfahren bestimmt werden können. Die Suche nach der Satzhypothese ist eine Suche nach einem Pfad mit minimalen Kosten durch diese Matrix, wobei aufgrund der physikalischen Natur der Sprachsignale Randbedingungen zu beachten sind: Man darf weder rückwärts gehen noch einzelne Phoneme endlos dehnen.

[4] Daher die in der Literatur verbreitete Bezeichnung *Dynamic Time Warping* (DTW).

Das Kernstück des *one-stage* Algorithmus läßt sich wie folgt formulieren:
Eingabe: Sprachsignal (von $t = 1, \ldots, N$), Wortmodelle , Sprachmodell
Ausgabe: Satzhypothese mit minimalen kumulierten Kosten

1. *Schritt:* Initialisierung
   Die erste Spalte der Matrix repräsentiert den Beginn des Sprachsignals. Sie wird für die gemäß Sprachmodell am Satzanfang zugelassenen Worte mit den Bewertungen für die Phoneme initialisiert, die gemäss Wortmodell am Wortanfang sind. Für alle anderen Wortmodelle wird sie dagegen auf einen „unendlich" großen Wert gesetzt.

2. *Schritt:* Durchschreite die DP–Matrix entlang der Zeitachse
   Für alle Wörter versuche eine Transition innerhalb desselben Wortes gemäß Wortmodell. Für all jene Wörter, deren „Ende" man erreicht hat, versuche eine Transition zu einem durch das Sprachmodell zugelassenen Nachfolgewort. Übergänge in ein Nachfolge–Wort werden durch Zeiger abgespeichert.

3. *Schritt:* Erstellen der Satzhypothese
   Zunächst wird der Gitterpunkt bestimmt, der am Ende des Sprachsignals minimale kumulierten Kosten aufweist. Von diesem Punkt aus werden die Übergänge zwischen Wörtern zurückverfolgt. Dabei werden die während des Durchschreitens der DP–Matrix bestimmten Zeiger an Wortübergängen verwendet.

Abb. 2.3 stellt den durch den *one-stage* Algorithmus bestimmten optimalen Pfad für den Beispielsatz „Mein Name ist Harald Bovic" graphisch dar.

Um die Suche effizient durchzuführen, werden im allgemeinen Strahlsuch–Methoden eingesetzt. Die zugrundeliegende Idee ist, die Suche auf einen begrenzten Bereich in der Nähe der Satzhypothese mit minimalen Kosten einzuschränken. Im *one-stage* Algorithmus kann eine Strahlsuche realisiert werden, indem eine Menge „aktiver" Wortmodelle eingeführt wird. Zu jedem Zeitpunkt des Durchschreitens der DP–Matrix werden die bis dahin minimalen kumulierten Kosten bestimmt. Transitionen innerhalb von Wortmodellen oder zu einem neuen Wortmodell werden nur dann durchgeführt, wenn die zugehörige kumulierten Kosten nicht über einem Schwellwert liegen. Ansonsten wird das Wortmodell deaktiviert. Durch geeignete Wahl des Schwellwertes kann man erreichen, daß der optimale Pfad zu jedem Zeitpunkt mit hoher Wahrscheinlichkeit innerhalb des Strahles liegt.

2.2.3 Algorithmen zur Bestimmung der N besten Satzhypothesen

Die in diesem Abschnitt vorgestellten Algorithmen sind Arbeiten von R. Schwartz und S. Austin [13, 14, 15] entnommen.

Ein Algorithmus zur Bestimmung der N besten Satzhypothesen kann durch einen abgeänderten *one-stage* Algorithmus erfolgen. Die Grundidee ist, Informationen über Teil–Satzhypothesen (Theorien) mit verschiedenen Wort–Vorgeschichten während des Durchschreitens der DP–Matrix abzuspeichern. Am Ende wird nicht nur die Hypothese mit minimalen kummulierten Kosten zurückverfolgt, sondern auch alternative Hypothesen mit höheren Kosten.

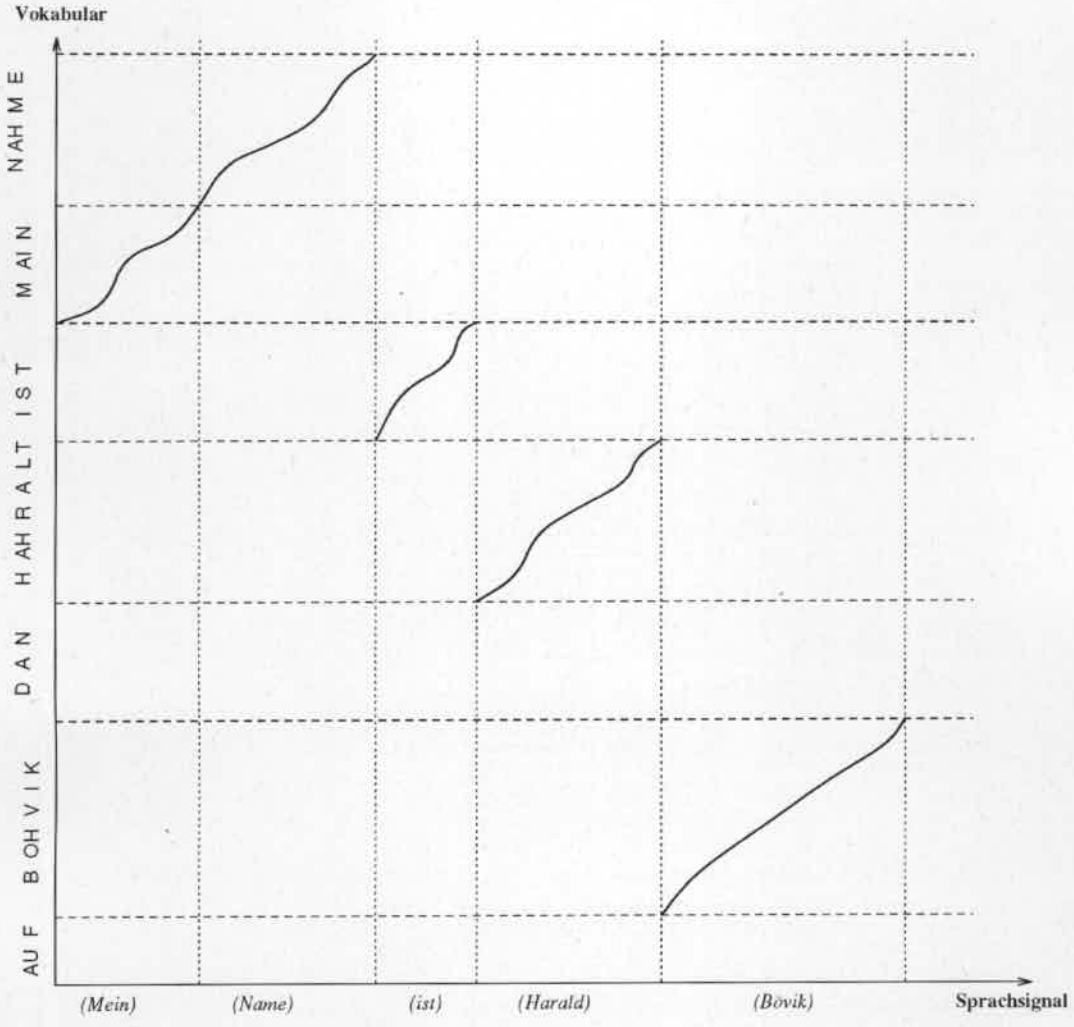

Abbildung 2.3: DP–Matrix für „Mein Name ist Harald Bovik“

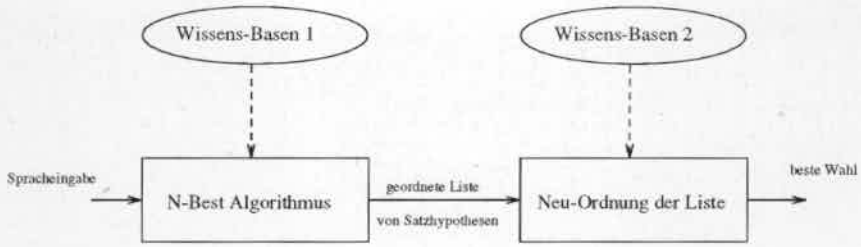

Abbildung 2.4: Das *N-best* Such-Paradigma

Hierzu wird jeder Pfad mit einem Zeiger auf seine kummulierten Kosten und auf seine Wort-Vorgeschichte markiert. Dabei ist unter der Wort-Vorgeschichte die Wortfolge vom Beginn des Sprachsignals bis zum aktuellen Zeitpunkt zu verstehen. Falls zwei Pfade im Verlauf der Suche zusammentreffen, werden sie nur dann getrennt fortgeführt, wenn ihre Vorgeschichten verschieden sind. Aus Effizienzgründen ist auch hier der Einsatz von Strahlsuch-Methoden notwendig: Theorien, deren kumulierte Kosten einen bestimmten Schwellwert oberhalb der optimalen Theorie liegen, scheiden aus der Suche aus.

Eine Beschleunigung läßt sich durch den *forward-backward* Algorithmus erreichen. Er erfordert zwei Such-Durchläufe: Der erste erfolgt in Vorwärts-Richtung und bestimmt nur die *first best* Hypothese. Dazu kann der *one-stage* Algorithmus verwendet werden. Dabei werden Informationen über die im Verlauf der Suche durchlaufenen Wortmodelle abgespeichert. Daran schließt sich eine zweite Phase an, die das Sprachsignal nochmals in Rückwärtsrichtung (vom Ende zum Anfang) durchläuft. In dieser Suche werden die N besten Satzhypothesen bestimmt. Entscheidend ist, daß nicht mehr der gesamte Suchraum durchlaufen wird, sondern nur ein eingegrenzter Bereich um den in der Vorwärts-Suche gefundenen optimalen Pfad. Dadurch wird die Laufzeit drastisch verkürzt.

Hinter der Anwendung von Algorithmen zur Bestimmung der N besten Satzhypothesen steht das folgende *N-best Such-Paradigma* (siehe Abbildung 2.4): Mit Hilfe effizienter und wirkungsvoller Wissensbasen (zum Beispiel statistische Grammatiken als Sprachmodell) wird eine Liste der N besten Satzhypothesen erzeugt. Die Satzhypothesen werden anschließend mittels weiterer Wissensbasen neu bewertet und geordnet, zum Beispiel Satzteilanalyse oder semantische Analyse. Diese Wissensbasen sind zwar für das Auffinden der korrekten Hypothese sehr hilfreich, können jedoch in der Suche nicht effizient implementiert werden. Abschließend wird die Hypothese mit der besten neuen Bewertung als Satzhypothese gewählt.

2.3 Statistische Grammatiken

Im folgenden werden Grundlagen statistischer Grammatiken und ihre Anwendung in Spracherkennern als Sprachmodell aus der Veröffentlichung von F. Jelinek [12] referiert. Es wird auch kurz der Begriff der Perplexität definiert. Er dient als Maß zur Bewertung von statistischen Grammatiken. In einem weiteren Teilabschnitt wird eine von S. Katz in [16] vorgestellte Methode dargestellt, statistische Grammatiken aus wenig umfangreichen Datenmengen zu generieren.

2.3.1 Grundlagen

Eine *statische Grammatik* weist jeder Wortfolge $W = w_1 w_2 \cdots w_n$ eine Wahrscheinlichkeit $P(W)$ zu. Die Gesamtwahrscheinlichkeit läßt sich hierbei sequentiell aus den Wahrscheinlichkeiten von Teil–Wortfolgen berechnen:

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_1, \ldots, w_{i-1}) \tag{2.1}$$

Die Wahrscheinlichkeiten werden in der Praxis anhand von Häufigkeiten von Wortfolgen in Texten geschätzt. Um zuverlässige Schätzwerte zu erhalten, identifiziert man häufig alle Wortfolgen, die in den N letzten Wörternübereinstimmen, mit dem **N–Gramm** $w_{i-N+1} \cdots w_i$. Die Formel zur Berechnung der Wahrscheinlichkeit eines Satzes nimmt dann für *Trigramme* ($N = 3$) die folgende Form an:

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) \tag{2.2}$$

Die Anwendung einer statistische Grammatik als Sprachmodell beruht auf der Tatsache, daß der Spracherkenner eine *Maximum–Likelihood* Entscheidung fällt: Er entscheidet sich für diejenige Satzhypothese $\hat{W}$, die bezüglich des Sprachsignals A am wahrscheinlichsten ist. Sie muß also folgende Gleichung erfüllen[5]:

$$P\left(\hat{W} \mid A\right) = max_W P(W \mid A) \tag{2.3}$$

Mit der Formel von Bayes für bedingte Wahrscheinlichkeiten läßt sich die auf der rechten Seite auftretende Wahrscheinlichkeit umformen zu

$$P(W \mid A) = \frac{P(W)\, P(A \mid W)}{P(A)} \tag{2.4}$$

wobei die Wahrscheinlichkeit $P(A \mid W)$ für das akustische Signal A bei bekannter Wortfolge aus den Phonemmodellen des Erkenners bestimmt werden kann. $P(W)$ wird durch das Sprachmodell festgelegt. $P(A)$ ist die Wahrscheinlichkeit für das akustische Signal. Diese bleibt bei der Maximierung gemäß Gleichung (2.3) konstant und braucht daher nicht weiter betrachtet zu werden.

2.3.2 Begriff der Perplexität

In Anwendung des Begriffes der Entropie auf statistische Grammatiken kann der Informationsgehalt eines Wortes aus einem ausreichend langen Satz $w_1 \cdots w_n$ geschätzt werden gemäß

$$H \approx -\frac{1}{n} \log P(w_1 \cdots w_n)$$

[5] $P(A \mid B)$ bezeichnet die Wahrscheinlichkeit des Ereignisses A unter der Bedingung, daß das Ereignis B bereits eingetreten ist.