Vertrauensmaße für die maschinelle Spracherkennung

Diplomarbeit

von

Thomas Schaaf

Bearbeitungszeitraum:

1. Mai 1996 - 31. Oktober 1996

Institut für Logik, Komplexität und Deduktionssysteme
Fakultät für Informatik
Universität Karlsruhe (TH)
D-76128 Karlsruhe

Betreuer:

Prof. Alexander Waibel
Dipl. Phys. Thomas Kemp

Hiermit erkläre ich, die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen verwendet zu haben.

Karlsruhe, 31. Oktober 1996

Thomas Schaaf

Abstrakt:

Spracherkenner produzieren Hypothesen mit einer hohen Wortakkuratheit. Sie bieten aber bisher normalerweise keine Bewertung der Glaubwürdigkeit der Worte der gefundenen Hypothese an. In dieser Arbeit wird für verschiedene Merkmale untersucht, wie nützlich sie für die Bewertung der Glaubwürdigkeit eines Wortes sind.

Basierend auf den gefundenen Merkmalen wurde eine Vektorklassifikator (Neuronales Netz) eingesetzt, um einen Vertrauensmesser zu realisieren. Verschiedene Merkmalskombinationen wurden untersucht, um den Beitrag der einzelnen Merkmale zur Klassifikation aufzuzeigen. Die beste Merkmalskombination hatte eine Fehlerreduktion von 36,9 % verglichen mit einem a priori Klassifikator.

Abstract:

Speech recognition systems produce hypothesises with a high word accuracy. But usually they don't give a measure of confidence of the words in the hypothesis. In this work we examine several features with respect to there usefulness in estimating word confidence.

Based on the found features a vector classifier (Neural Network) was used to build a confidence measure system. Different feature combinations were examined to show the contribution of the features to the classifier. The best feature combination has an error reduction about 36,9 % compared with an a priori classifier approach.

Danksagung:

Bedanken möchte ich mich bei Thomas Kemp, der mich für das Thema der Arbeit begeistert hat, und von dem ich sehr viel lernen konnte. Weiter geht mein Dank an Prof. Waibel für die Möglichkeit, diese Arbeit an seinem Institut durchführen zu können. Ein Dank auch an Michael Finke für sein Interesse und die nützlichen Hinweise, die diese Arbeit sehr bereichert haben. Es bleibt noch all den Mitarbeitern des Lehrstuhls zu danken, da sie sich stets Zeit genommen haben, Fragen über den Umgang mit Janus-3 zu klären.

Inhaltsverzeichnis

1 Einleitung **9**

1.1 Inhaltsübersicht 12

2 Grundlagen **13**

2.1 Sprache und Sprachklassen bei der maschinellen Spracherkennung 13

2.2 Datensammlung 14

2.3 Statistischer Ansatz zur Spracherkennung 16

2.4 Vorverarbeitung 17

2.5 Markov-Modelle für die Spracherkennung 18

2.6 Viterbi-Algorithmus 20

2.7 Das Sprachmodell 21

2.8 Training der akustischen Modelle 22

2.9 Der Worthypothesengraph 24

2.10 Entropie 27

3 Bewertung von Vertrauensmessern **33**

3.1 Der Align-Algorithmus 33

3.2 Einteilung von Vertrauensmessern 36

3.3 Qualitätsmaße für Vertrauensmesser 38

4 Der Spracherkenner und die Datenbasis **47**

4.1 Janus-2-System und Janus-3-Programm 47

4.2 Verbmobil-Datenbasis 48

5 Untersuchte Merkmale **53**
5.1 Wortassoziierte Merkmale . . . 53
5.1.1 Wortidentität . . . 54
5.1.2 Wortlänge . . . 56
5.1.3 Sprachmodell . . . 57
5.1.4 Häufigkeit im akustischen Training . . . 58
5.1.5 Bewertung der wortassoziierten Merkmale . . . 60
5.2 Zur Sprechgeschwindigkeit assoziierte Merkmale . . . 61
5.2.1 Wortstreckung und Wortstauchung . . . 61
5.2.2 Schwankung der estimierten Sprechgeschwindigkeit . . . 64
5.2.3 Bewertung der zur Sprechgeschwindigkeit assoziierten Merkmale . . . 65
5.3 Akustische Ähnlichkeit . . . 66
5.3.1 Mittlere Wort-Score . . . 67
5.3.2 Viterbi-normalisierte Wort-Score . . . 67
5.3.3 A priori normalisierte Wort-Score . . . 68
5.3.4 Erwartete mittlere Wort-Score . . . 69
5.3.5 Bewertung der akustischen Ähnlichkeit . . . 70
5.4 Unsicherheit im Suchraum . . . 71
5.4.1 Akustische Stabilität . . . 72
5.4.2 Entropie im Worthypothesengraph . . . 73
5.4.3 Bewertung der Unsicherheit im Suchraum . . . 77

6 Experimente und Ergebnisse **79**
6.1 Der Klassifikator . . . 79
6.2 Merkmalkombinationen . . . 80
6.3 Ergebnisse . . . 81

7 Zusammenfassung und Ausblick **85**
7.1 Zusammenfassung . . . 85
7.2 Ausblick . . . 88

A Berechnung der LatEntropie **95**

B Zusammenfassung: Korrelationen **103**

Tabellenverzeichnis

2.1 Attribute eines Knotens 27
2.2 Attribute einer Kante 28

3.1 Fehlerzuordnung 1 34
3.2 Fehlerzuordnung 2 34
3.3 Kosten der Operationen bei Editierdistanz 35
3.4 Verwendete Notation 38

4.1 Datenmengen: Aufteilung nach Sammelorten 49
4.2 Datenmengen: Aufnahmedauer, Sprechergeschlecht, Worte insgesamt 50
4.3 Datenmengen: Korrektraten in den Hypothesen 50
4.4 Aufteilung der Fehler in den Datenmengen 51

5.1 Fehlerrate über die Wortidentität 55
5.2 Korrelation: Wortidentität 55
5.3 Bewertung des Klassifikators: Wortidentität 56
5.4 Korrelation: Wortlänge 56
5.5 Korrelation: Sprachmodell 58
5.6 Fehlerrate: Sprachmodell 58
5.7 Korrelation: Häufigkeit im akustischen Training 59
5.8 Korrelation: Wortassoziierte Merkmale und Fehler 60
5.9 Korrelation: Wortassoziierte Merkmale untereinander 60
5.10 Korrelation: Wortstreckung und Wortstauchung 63

5.11 Korrelation: Wortschwankung und Kontextschwankung 65
5.12 Korrelation: Assoziierte Merkmale der Sprechgeschwindigkeit untereinander 66
5.13 Korrelation: (normalisierte) Wort-Score 70
5.14 Korrelation: akustische Ähnlichkeit 71
5.15 Korrelation: akustische Stabilität 73
5.16 Parameterkombinationen mit deutlicher Korrelation zur Fehlerrate 75
5.17 Korrelation: Entropie 75
5.18 Korrelation: Unsicherheit im Suchraum 78

6.1 Korrelation: Ausgewählte Merkmale 81
6.2 Korrelation: Ausgewählte Merkmale untereinander (Teil 1) 82
6.3 Korrelation: Ausgewählte Merkmale untereinander (Teil 2) 82
6.4 Untersuchte Merkmalskombinationen 83
6.5 Klassifikationsergebnisse 84
6.6 Fehlerreduktion des besten Klassifikators 84

A.1 Kantentypen 96
A.2 Knotentypen 96
A.3 Berechnungsmodi für die Kantenscore 99

B.1 Zusammenfassung der Korrelationen 103

Abbildungsverzeichnis

2.1 Zusammenhang Sprachprobe, Aufnahmen, Äußerung und Transkriptionen 16

2.2 Aufbau eines Spracherkenners 17

2.3 mel-Scale Filterbank 18

2.4 Einfaches HMM 19

2.5 Mögliche HMM-Realisierungen für Worte/Phoneme 19

2.6 Suchraum mit Viterbi-Pfaden 25

2.7 Darstellung der Viterbi-Pfade als Graph 26

2.8 Die resultierende Lattice 26

2.9 Verbindungsstruktur der Lattice–Elemente 26

2.10 Umwandlung einer Mehrfachentscheidung 28

2.11 Gerichteter Graph mit Übergangswahrscheinlichkeiten 30

2.12 Baum mit Übergangswahrscheinlichkeiten 30

2.13 Vereinfachter Baum mit Übergangswahrscheinlichkeiten 31

2.14 Struktur mit exponentieller Anzahl von Pfaden 31

3.1 Arbeitsweise des Align-Algorithmus 35

3.2 Zeitliche Zuordnung von Referenz- und Hypothesensatz 37

3.3 Kommunikationskanal 40

3.4 Binärer Kommunikationskanal 44

3.5 Präzision und Ausschöpfung für die Klasse korrekt über der Schwelle (links) und gegeneinander (rechts) 45

5.1 Gesamtfehler (links) und Einfügefehler (rechts) über die logarithmierte Wortlänge . . . 57
5.2 Gesamtfehler (links) und OOV-Fehler (rechts) über logarithmische Häufigkeit im akustischen Training . . . 59
5.3 Gesamtfehler (links) und OOV-Fehler (rechts, nur WStauch3) über die Wortstauchung . . . 63
5.4 Gesamtfehler über die Schwankung der Sprechgeschwindigkeit (links WSchw, rechts KSchw) . . . 65
5.5 Gesamtfehler (links) und Einfügefehler (rechts) MWScore . . . 70
5.6 Gesamtfehler über AMWScore (links) und EMWScore (rechts) . . . 71
5.7 Gesamtfehler über akustischer Stabilität (AStabil) . . . 73
5.8 Gesamtfehler über H1, H2 (links) und H3 (rechts) . . . 76
5.9 Einfügefehler über H1, H2 (links) und H3 (rechts) . . . 76
5.10 OOV-Fehler über H1, H2 (links) und H3 (rechts) . . . 77

6.1 Eingesetzte Netzstruktur . . . 79

A.1 Lattice L mit Zeitbereichen und Kantentypen . . . 96
A.2 Teilgraph L' der Lattice L . . . 98
A.3 Lattice mit lokalen Wahrscheinlichkeiten . . . 101
A.4 Normalisierte Lattice in/gegen Zeitrichtung (links/rechts) . . . 101
A.5 Baum-Welch: Zwischenergebnis (links) und Ergebnis (rechts) . . . 102
A.6 Normalisierte Lattice aus Baum-Welch in/gegen Zeitrichtung (links/rechts) . . . 102

Kapitel 1

Einleitung

"Und wenn Ihr Euch nur selbst vertraut,
Vertrauen Euch die andern Seelen."
Mephistopheles in Faust [2021/2022], Goethe.

Spracherkenner haben einen Reifegrad erreicht, der einen Einsatz im alltäglichen Leben nicht mehr als reine Vision erscheinen läßt, sondern bereits Realität angenommen hat. Beispiele sind Diktier- und Auskunftsysteme.

Die erfolgreichsten Spracherkenner setzen statistische Methoden der Mustererkennung ein, die zu einer akustischen Eingabe eine Hypothese produzieren, die vom Standpunkt der Wahrscheinlichkeitstheorie die glaubwürdigste ist, da sie am wahrscheinlichsten ist. Aber selbst die wahrscheinlichste Hypothese kann Fehler enthalten und es wäre wünschenswert, daß der Spracherkenner jedes einzelne Wort seiner Hypothese mit einem Vertrauensmaß bewertet. Hat ein Spracherkenner beispielsweise die Hypothese «Ja ich möchte zehn Millionen Brötchen» gefunden, gesagt wurde aber «Ich möchte zehn Milchbrötchen», so würde ein *perfekter* Vertrauensmesser dazu die Ausgabe «ja» = 0 %, «ich» = 100 %, «möchte» = 100 %, «zehn» = 100 %, «Millionen» = 0 % und «Brötchen» = 0 % liefern. Ein Wert von 100 % bedeutet, daß das Wort ganz sicher korrekt und ein Wert von 0 %, daß das Wort ganz sicher falsch erkannt wurde. Ein *guter* Vertrauensmesser sollte so nahe wie möglich an die perfekte Bewertung kommen. Dabei soll ein großer Wert, beispielsweise 85 % andeuten, daß das Wort eher korrekt als falsch erkannt ist und ein kleiner Wert, beispielsweise 15 %, auf einen Erkennungsfehler hinweisen.

Anwendungsmöglichkeiten für einen Vertrauensmesser

Basierend auf den Ausgaben des Vertrauensmessers können neue Techniken in der Spracherkennung entwickelt und bestehende verbessert werden. Verschiedene Anwendungsmöglichkeiten sind:

- Ablehnung einer Hypothese wenn nicht genügend Vertrauen in diese vorliegt.
- Gezieltes Nachfragen bei einem vermutlichen Fehlerkennen zur Verbesserung des Mensch-Maschine-Dialogs und der Versuch, aufgrund des richtig verstandenen Teils der Hypothese eine sinnvolle Einschränkung der Frage zu erreichen. Beispielsweise nicht ≪Können Sie bitte alles noch einmal wiederholen?≫, sondern Teilverstehen durch den Parser, der dazu die notwendigen Hinweise erhalten muß, und Nachfragen der unsicheren Zusammenhänge. Beispielsweise ≪Ich habe den Wochentag nicht verstanden, können Sie ihn bitte wiederholen?≫
- Beschleunigung der Suche, indem mit einem vereinfachten Modell eine Hypothese berechnet, und bei Worten, die ein zu geringes Vertrauen erhalten haben, eine verbesserte Suche mit verfeinerten Modellen durchgeführt wird. Hierdurch kann möglicherweise Geschwindigkeit gegen Genauigkeit abgewogen werden, beispielsweise wenn ein Wort, das möglicherweise falsch erkannt wurde, eine wichtige Bedeutung haben kann.
- Analyse einfach zu erkennender Fehler[1], wodurch möglicherweise ein Einblick in die Fehlerursache zu gewinnen ist. Dabei können vielleicht neue Erkenntnisse gewonnen werden, die zur Verbesserung des Spracherkenners führen.
- Durchführung automatischer Transkription von Trainingsdaten, bei denen, nur noch wenn nötig, die Hypothese von einem Menschen validiert wird. Dadurch kann aufwendige und kostenintensive Arbeit eingespart werden. Wenn nur auf den Teilen, die ein großes Vertrauen besitzen, ein akustisches Training durchgeführt wird, kann möglicherweise ein Spracherkenner bereits mit einer verhältnismäßig kleinen Menge an transkribierten Äußerungen trainiert werden.
- Sprecheradaption (MLLR = Maximum Likelihood Linear Regression) ist auf einer korrekten Transkription am erfolgreichsten. Diese liegt aber im normalen Betrieb eines Spracherkenners nicht vor. Wird dagegen eine Sprecheradaption auf der gesamten (fehlerhaften) Hypothese durchgeführt ist

[1] Worte mit sehr geringem Vertrauen, die dann auch tatsächlich falsch sind.

der Erfolg nur noch gering. Eine Adaption auf dem überwiegend korrekten Anteil der Hypothese führt hingegen meist zu einer Verbesserung [8].

- Verwendung einer Kombination zweier unterschiedlicher Spracherkenner, die gleichzeitig eingesetzt werden. Anhand des Vertrauensmessers wird entschieden welche der erzeugten Hypothesen ausgegeben werden soll.

- Verwendung einer Kombination von akustischen und visuellen Signalen (Lippenlesen). Ein Vertrauensmesser kann beispielsweise dann, wenn sich der akustische Spracherkenner unsicher ist, den visuellen Teil stärker gewichten (Sensorfusion) [10], beispielsweise in Umgebungen mit vielen Störgeräuschen.

- Einbeziehung neuer Wissensquellen durch den Vertrauensmesser, die nicht (optimal) vom Spracherkenner ausgenutzt werden können. So besteht die Möglichkeit, daß durch das Umbewerten von N-Besten-Listen oder Worthypothesengraphen die Erkennungsleistung verbessert werden kann.

Theoretische Grenzen eines Vertrauensmessers

Viele der hier genannten Anwendungen negieren die eigentliche Aufgabe eines Vertrauensmessers, da sie eingesetzt werden, um eine bessere Hypothese zu erzeugen und danach keine Aussage über das Vertrauen in die neue Hypothese möglich ist. Es würde ein Vertrauensmesser benötigt, der für so erzeugte Hypothesen zuständig ist. Eine unendliche Schleife zur Verbesserung der Erkennungsleistung ist dabei aber nicht möglich, denn es würde wieder eine nicht optimal ausgenutzte Wissensquelle benötigt. Wird davon ausgegangen, daß bereits beim ersten Schritt alle vorhandenen Wissensquellen einbezogen wurden, kann nur noch die Glaubwürdigkeit bewertet werden. Die Bewertung erscheint deswegen möglich, da für eine Bewertung der Glaubwürdigkeit nicht entschieden werden muß, welches Wort ausgegeben wird, sondern ob eine Verwechslungsgefahr besteht. Dies ist im Vergleich zur Spracherkennung eine ganz andere Aufgabe.

Vorgehensweise

Um nun einen Vertrauensmesser für die maschinelle Spracherkennung zu erhalten, ist es zunächst notwendig Merkmale zu finden, die darauf hinweisen, ob ein Wort eher korrekt oder fehlerhaft erkannt wurde. Diese Merkmale sind für jedes Wort der Hypothese zu bestimmen. Daraus ergibt sich dann zu jedem Wort ein Merkmalsvektor. Mit einem Vektorklassifikator wird dann untersucht, wie wahrscheinlich ein korrekt erkanntes Wort vorliegt.

Ziel dieser Arbeit ist es einen Vertrauensmesser für den Janus-3-Spracherkenner zu realisieren und zu untersuchen.

1.1 Inhaltsübersicht

Kapitel 2 enthält Grundlagen über Spracherkennung und legt die in der Arbeit verwendeten Begriffe fest. Außerdem wird der Aufbau eines Worthypothesengraphen des Janus-3-Programms beschrieben und die Eigenschaften der Entropie als Maß für Unsicherheit dargelegt.

In Kapitel 3 wird eine Einteilung der Erkennungsfehler mit Hilfe des Align-Algorithmus vorgenommen. Es folgt eine Einteilung für Vertrauensmesser basierend auf deren Ausgabewert. Abschließend werden verschiedene Qualitätsmaße zur Bewertung von Vertrauensmesser untersucht und diskutiert.

In Kapitel 4 wird zunächst der verwendete Spracherkenner (Janus-2-System) vorgestellt. Danach wird die Datenbasis beschrieben, die für die Bewertung der untersuchten Merkmale, das Training und die Bewertung des Vertrauensmessers verwendet wurde.

Kapitel 5 beschreibt die untersuchten Merkmale, die für eine Bewertung der Glaubwürdigkeit aussichtsreich erscheinen. Die Merkmale wurden in vier Gruppen unterteilt, von denen jede in einem Abschnitt behandelt wird.

In Kapitel 6 wird das zur Klassifikation durchgeführte Experiment beschrieben. Anschließend wird untersucht, welchen Beitrag die einzelnen Merkmalskombinationen zur Qualität des Vertrauensmessers leisten.

Kapitel 7 gibt eine kurze Zusammenfassung der Arbeit und der Ergebnisse. Offengebliebene Fragen und ein Ausblick auf zukünftige Arbeiten bilden den Abschluß.

Im Anhang A befindet sich eine Beschreibung des im Rahmen dieser Arbeit entwickelten Merkmals LatEntropie. Es werden Schwierigkeiten diskutiert, die sich aus der vereinfachten Berechnungsvorschrift ergeben und dargelegt, wie diese gelöst wurden.

Kapitel 2

Grundlagen

In diesem Kapitel wird zunächst eine Einteilung von Sprachklassen für die maschinelle Spracherkennung gegeben und anschließend erläutert, wie eine Sprachprobe gewonnen wird. Es wird eine kurze Beschreibung der Aufgabe der Vorverarbeitung in der Spracherkennung gegeben. Danach werden kurz die für die Spracherkennung wichtigen Markov-Modelle betrachtet. Eine Beschreibung der Arbeitsweise eines Spracherkenners wird im Abschnitt über den Viterbi-Algorithmus gegeben. Es folgt eine kurze Erklärung über Sprachmodelle und die Aufgabe des akustischen Trainings. Wie im Spracherkenner Janus-3 eine Menge von möglichen Hypothesen dargestellt wird, beschreibt der Abschnitt über den Worthypothesengraphen. Als Abschluß des Kapitels wird die Möglichkeit, Unsicherheit als Entropie aufzufassen dargelegt und erklärt, wie diese in einem Graphen effizient berechnet werden kann.

2.1 Sprache und Sprachklassen bei der maschinellen Spracherkennung

Sprache dient der Kommunikation und kann in verschiedener Form auftreten. Man unterscheidet geschriebene Sprache, wie beispielsweise diesen Text, und gesprochene Sprache. Taubstumme Menschen bedienen sich einer Gebärdensprache und können teilweise Sprache von den Lippen ablesen. Hier soll gesprochene Sprache betrachtet werden. Gesprochene Sprache besteht aus Klang und Phonemfolgen. Beim Menschen wird sie durch die Erregung der Stimmbändern und die Stellung des Vokaltraktes erzeugt. Begleitet wird sie von Ausdrucksmitteln wie Tonhöhe, Lautstärke und Tonrhythmus und nichtverbalen Ausdrucksmitteln wie beispielsweise Mimik und Gestik.

Bei der maschinellen Spracherkennung kann Sprache nach verschiedenen Kriterien eingeteilt werden, die am Schwierigkeitsgrad bei der Erkennung orientiert sind, und die sich in den Einschränkungen äußern, die dem Sprecher auferlegt werden.

Am einfachsten maschinell zu erkennen ist ein Sprachsystem aus *Einzelworten*. Jede Äußerung darf hierbei nur ein einziges Wort enthalten. Üblicherweise verfügt ein solches System nur über ein kleines Vokabular. Beispiel hierfür ist ein Ziffernerkenner, dessen Vokabular aus den zehn Ziffern und den Worten ≪ja≫ und ≪nein≫ besteht. Die Anwendungsmöglichkeiten für solche Systeme sind umfangreich, beispielsweise läßt sich ein menügeführtes Auskunftssystem realisieren.

Bei *semikontinuierlicher Sprache* muß vom Sprecher nach jedem Wort eine deutliche Pause eingefügt werden, um eine Segmentierung der Äußerung in einzelne Worte zu unterstützen. Diese Systeme erfordern vom Anwender neben einer Anlernphase eine hohe Disziplin beim Sprechen. Für solche Sprachen existieren Erkennungssysteme, die mit einem großem Vokabular von 20000 Worten bereits für Diktiersysteme eingesetzt werden.

Bei *kontinuierlicher Sprache* entfällt die künstliche Pause zwischen den Worten. Unterschieden wird zwischen abgelesener und frei gesprochener Sprache. Gelesene Sprache, beispielsweise Nachrichten im Radio oder Fernsehen, ist einfacher zu erkennen, da der Sprecher hier im allgemeinen flüssiger spricht. Spontane Sprache dagegen enthält oftmals Satzabbrüche und sogenannte 'Fehlstarts', bei denen mitten in einem Wort abgebrochen und (ohne Übergang) mit einem anderen Wort begonnen wird. Die Satzstrukturen unterscheiden sich deutlich von den Grammatikregeln für geschriebene Sprache und häufig werden vom Sprecher Pausenfüller, wie beispielsweise ≪ähm≫ oder ≪eh≫ verwendet. Zur Illustration diene das Beispiel: ≪Ich hab' ähm am Mittw{och} nein nächste Woche habe ich keine Zeit.≫ Auch hier kann eine Einschränkung gemacht werden. Da spontan gesprochene Sprache aus einem nahezu unbegrenzten Wortschatz schöpfen kann, wird aktuell bei der Anwendung das Gesprächsthema beschränkt. Es sind zwar weiterhin alle Worte theoretisch erlaubt, aber die Beschränkung auf eine Domäne grenzt das zu erwartende Vokabular ein.

2.2 Datensammlung

Die Datensammlung ist eine wichtige Vorarbeit. Je größer die Menge der verfügbaren Daten, desto besser können statistische Verfahren für die Spracherkennung eingesetzt werden. Um eine Sprachprobe zu erhalten werden folgende Schritte durchgeführt:

1. Motivation: durch Werbung oder direktes Ansprechen werden Spender geworben. Dabei müssen bereits Hemmschwellen abgebaut werden, beispielsweise wegen der Speicherung der Identität.

2. Instruktion: dem Spender wird die technische Ausrüstung, die Bedienung des Aufnahmeprogramms und das Szenario, dem die Spende unterliegt, erklärt. Dabei sollen Hemmschwellen abgebaut werden, so daß der Spender sich entspannt und eine für ihn typische (natürliche) Sprachprobe geben kann. Die (künstliche) Laborbedingung soll der Spende möglichst nicht angemerkt werden.

3. Aufnahme: die Spender führen eine dem Szenario angemessene Spende durch, beispielsweise bei *Verbmobil*[1] eine Terminabsprache für einem bestimmten Zeitraum basierend auf vorgegebenen Kalendern. Die Sprachspende wird digital aufgezeichnet, wozu qualitativ hochwertige analog-digital Konverter mit hochwertigen Mikrofonen verwendet werden. Möglich ist es auch einen DAT-Rekorder für die Aufnahme einzusetzen.[2]

4. Transkription: bei der Transkription wird eine Sprachprobe nach festgelegten Transkriptionsregeln niedergeschrieben. Dabei wird festgehalten was tatsächlich gesagt wurde, das bedeutet insbesondere Aussprachefehler, Dialektäußerungen, Wortabbrüche, etc. Zusätzlich werden alle auffälligen Geräusche vermerkt. Häufige menschliche (Stör-)Geräusche sind Einatmen, Zungenschlag, Lachen oder Hintergrundgemurmel. Nicht vom menschlichen Stimmapparat erzeugte Störgeräusche sind beispielsweise Türenschlagen, Telefonklingeln oder das Mausklicken beim Bedienen des Aufnahmeprogramms. Nicht zuordenbare und seltene Geräusche, wie beispielsweise ein vorbeifliegender Helikopter, sind in einer Restklasse zusammengefaßt.

5. Qualitätskontrolle: jede Sprachprobe wird von einer zweiten Person mit der Transkription verglichen, um Zweifel über die Bedeutung schwierig zu verstehender Textstellen oder die Geräuschzuordnung auszuräumen. Hierdurch wird erreicht, daß die Geräuschklassen möglichst konsistent sind und die Transkription allgemein weniger Fehler enthält.

6. Freigabe: sind mögliche Fehler ausgeräumt, wird die Sprachprobe mit der Transkription freigegeben.

Nach der Freigabe steht eine Sprachprobe zur Verfügung, die aus verschieden langen Äußerungen (Sprachsegmente, Utterances) und der Transkription besteht (Abbildung 2.1). Die *Aufnahme* eines Sprechers enthält mehrere *Äußerungen*, die wiederum aus mehreren Sätzen bestehen können.

[1]Verbmobil ist ein Projekt des Bundesministeriums für Forschung und Technologie (BMFT) mit dem Ziel ein System zu entwickeln, das die Übersetzung eines Dialogs mit fremdsprachigen Gespächspartnern unterstützt.

[2]Einen Spendenaufbau mit DAT-Rekorder wird in [2] beschrieben.

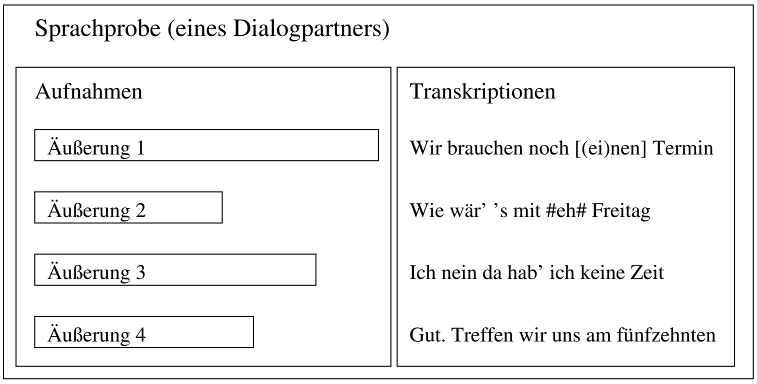


Abbildung 2.1: Zusammenhang Sprachprobe, Aufnahmen, Äußerung und Transkriptionen

2.3 Statistischer Ansatz zur Spracherkennung

Heutige Spracherkenner basieren auf Methoden der statistischen Mustererkennung. Die Aufgabe dabei ist eine Hypothese $\hat{W}$ zu einer Äußerung A zu finden, wobei die wahrscheinlichste Wortfolge zu bestimmen ist, also $\hat{W} = \arg\max_W p(W|A)$. Mit der Bayes-Formel

$$p(W|A) = \frac{p(A|W) * p(W)}{p(A)} \tag{2.1}$$

läßt sich diese Aufgabe umformulieren woraus sich nun die Hypothese durch

$$\hat{W} = \arg\max_W \frac{p(A|W) * p(W)}{p(A)} \tag{2.2}$$

bestimmen läßt.

Der Vorteil ist, daß für die Wortfolgen W einfache statistische Modelle eingesetzt werden können, um $p(A|W)$ zu berechnen. Diese Modelle bilden die akustischen Modelle eines Spracherkenners. Die a priori Wahrscheinlichkeit $p(W)$ wird dagegen von den Sprachmodellen eines Spracherkenners berechnet. Da die a priori Wahrscheinlichkeit für das Auftreten der Äußerung A ($p(A)$) zu einer gegebenen Äußerung unabhängig von der Wortfolge W ist, besitzt $p(A)$ keinen Einfluß auf die Maximumbildung und wird für die Bestimmung der Hypothese $\hat{W}$ nicht benötigt.

Abbildung 2.2 zeigt den Grundaufbau eines Spracherkenners mit seinen Modellen. Die Vorverarbeitung dient dabei der Merkmalsberechnung für die akustischen Modelle und in der Suche wird $\hat{W} = \arg\max_W p(A|W) * p(W)$ bestimmt. $\hat{W}$ ist dann die Hypothese des Erkenners.

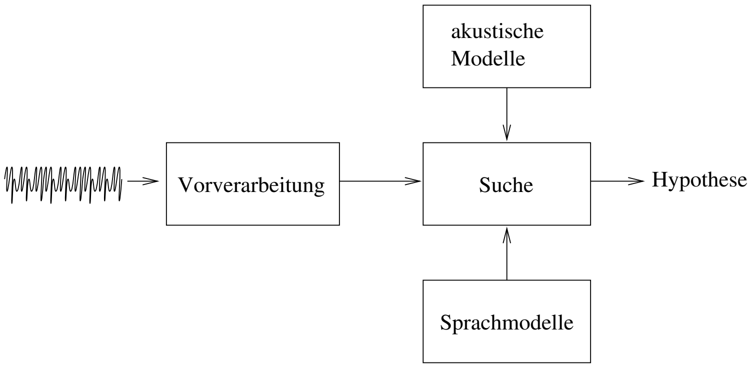


Abbildung 2.2: Aufbau eines Spracherkenners

2.4 Vorverarbeitung

Eine ausführliche Beschreibung, wie Signalvorverarbeitung für die Spracherkennung durchgeführt werden kann, ist in [24] und [38] zu finden. Hier folgt nur eine kurze Beschreibung welche Aufgaben die Vorverarbeitung hat.

Die Vorverarbeitung dient dazu, die wesentlichen Merkmale der Sprache hervorzuheben, störendes Rauschen zu unterdrücken und das Sprachsignal in eine normalisierte Darstellung zu übertragen, in der sprecherbedingte Unterschiede ausgeglichen sind. Beispielsweise besitzen verschiedene Sprecher üblicherweise einen verschieden langen Vokaltrakt, was das Sprachsignal beeinflußt. Zusätzlich wird eine Datenkompression angestrebt, um die Menge der zu verarbeitenden Daten gering zu halten.

Welche Merkmale muß nun eine Signalvorverarbeitung in der Spracherkennung hervorheben? Die wesentlichen Merkmale, an denen erkennbar ist, welches Wort gesagt wurde, sind in der zeitlichen Veränderung der Energie des Frequenzspektrums enthalten. Das menschlichen Gehörs besitzt im niederen Frequenzbereich eine wesentlich höhere Auflösung der Frequenzbänder als in den hohen Frequenzbereichen. Daraus läßt sich bereits eine verbreitete Methode ableiten mit der das Sprachsignal repräsentiert wird, die *Melscale-Vektoren* [25] [33]. Formel 2.3 beschreibt, wie Frequenz-Spektren in mel-Spektren umgerechnet werden. Das mel-Spektrum formt dabei die Frequenzen nicht linear um.

$$m = 1125 * \log(0.0016 * f + 1) \tag{2.3}$$

Wird nun die Energie der Frequenz f beispielsweise mit Dreiecksfiltern zusammengefaßt, die im mel-Spektrum äquidistante Bandbreite haben, so erhält man einen mel-Vektor. In Abbildung 2.3 ist eine Filterbank für 16 Frequenzgruppen abgebildet. Aus dieser Abbildung ist auch zu ersehen, wie die Auflösung im niederen Frequenzbereich deutlich feiner ist als für hohe Frequenzen.

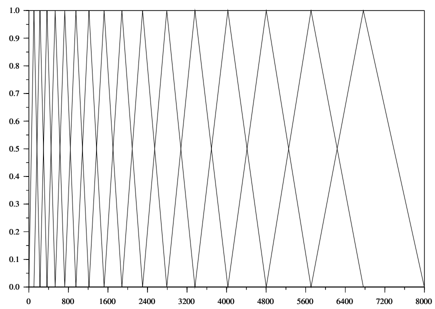


Abbildung 2.3: mel-Scale Filterbank

2.5 Markov-Modelle für die Spracherkennung

Sprache wird normalerweise als eine zeitliche Abfolge von Lauten (Phon) wahrgenommen. Dabei ist die *Reihenfolge* einzelner Phone wichtig für die Bedeutung eines Wortes, beispielsweise wie bei den Worten «mit» und «Tim». Modelle der Mustererkennung, bei denen eine zeitliche Reihenfolge berücksichtigt wird, sind *Versteckte Markov-Modelle* (HMM = Hidden Markov-Models) [22] [33], bei denen in jedem Zustand Observationen mit zustandsabhängigen Verteilungen möglich sind. In der Spracherkennung modellieren Zustände häufig Teile eines Wortes, beispielsweise ein Phonem, wovon wir im folgenden ausgehen.

Ein *HMM* besteht aus fünf Mengen und läßt sich durch ein 5-Tupel $\lambda = (S, K, O, P, \pi)$ charakterisieren:

1. Menge S von Zuständen $\{s_1 \dots s_N\}$
2. Menge K von gerichteten, gewichteten Kanten $k = (k_i, k_j, w_{i,j})$
3. Observationsalphabet O
4. Menge P von Emissionswahrscheinlichkeiten $P(O|j)$, $o \in O, 1 \leq j \leq N$
5. Initiale Zustandsbelegung $\pi = \pi_j, 1 \leq j \leq N$ der Zustände

Abbildung 2.4 zeigt ein einfaches HMM mit zwei Zuständen s_1 und s_2 in dem die Zeichen A und B des Observationsalphabet emitiert werden können.

Ein *HMM* wird in der Spracherkennung nach bestimmten Modelleigenschaften unterschieden. Eine Eigenschaft ist, wie viele Zustände verwendet werden, um ein Phonem zu modellieren. Möglich ist, für jedes Phonem genau einen Zustand zu verwenden, üblich sind aber drei oder mehr. Bei drei Zuständen kann der erste als

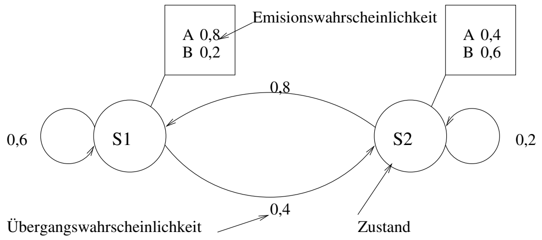


Abbildung 2.4: Einfaches HMM

Anlaufphase des Phonems angesehen werden, die noch durch die Nachbarschaft eines Vorgänger-Phonems beeinflußt ist. Auf die gleiche Weise läßt sich der dritte Zustand interpretieren, der das Ende des Phonems markiert, der bereits vom nachfolgenden Phonem beeinflußt wird. Der mittlere Zustand dagegen wird als vom Kontext unbeeinflußt angesehen. Diese speziellen Positionen werden hier mit 'a', 'm', 'e' für Anfang, Mitte und Ende abgekürzt. Abbildung 2.5 zeigt die verschiedenen Möglichkeiten.

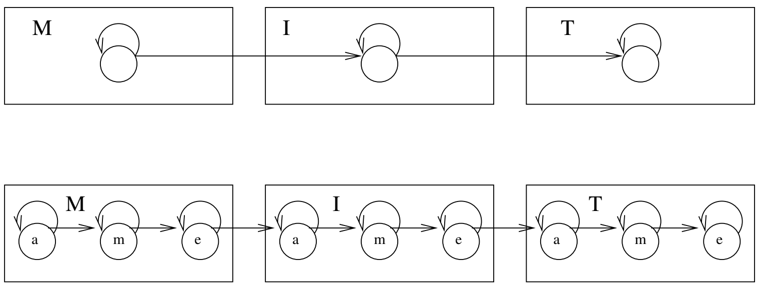


Abbildung 2.5: Mögliche HMM-Realisierungen für Worte/Phoneme

Wenn es in einem *HMM* zu den Phonemen unterschiedliche, von den Nachbarphonemen abhängende Modelle gibt, so wird ein solches Modell *kontextabhängiges HMM* genannt. Gibt es dagegen zu jedem Phonem nur ein einziges Modell, nennt man es ein *kontextunabhängiges HMM*. Beispielsweise kann in einem *kontextabhängigen HMM* der Zustand 'a' des Phonems ein anderer sein, je nachdem ob der linke Kontext ein Plosiv- oder ein Frikativlaut ist.

Da bei den hier vorgestellten Modellen nur auf die akustischen Eigenschaften der Worte eingegangen wird, werden diese Modelle *akustische Wortmodelle (akustische Modellierung)* genannt. Die Kombination von Wortmodellen wird mit Hilfe eines Sprachmodells durchgeführt (vgl. Abschnitt 2.7).

Im Zusammenhang mit einem HMM λ werden drei grundlegende Probleme unterschieden:

- Das Optimierungsproblem
  Wie mit einer (hinreichend langen) Beobachtung O die Modellparameter

so geschätzt werden können, daß das Modell besser zu den Beobachtungen paßt.

- Das Evaluationsproblem
  Wie wahrscheinlich es ist, daß eine gegebene Merkmalsfolge O von Modell λ erzeugt wird.

- Das Decodierungsproblem
  Wie zu einer Beobachtungsfolge O und einem Modell λ die wahrscheinlichste Zustandsfolge in λ gefunden wird, die O produziert.

Für die Lösung der Probleme gibt es verschiedene Algorithmen, die in [22] beschrieben sind. Wir wollen im folgenden Abschnitt das Decodierungsproblem näher betrachten.

2.6 Viterbi-Algorithmus

Die Aufgabe eines Spracherkenners ist, herauszufinden, was in einer Äußerung gesagt wurde. Dies wird als Decodierung und der Teil des Spracherkenners, der dies leistet, als Decoder bezeichnet. Hierzu wird ein HMM λ verwendet, das aus Wortmodellen besteht, die untereinander so verbunden werden, daß die erlaubten Wortfolgen möglich sind (vgl. Abschnitt 2.7).

Mit diesem Modell ist es nun anhand der Bayes-Formel 2.1 möglich, zu einer Observation O (Äußerung) für die möglichen Wortfolgen $W = w_1, \ldots, w_n$ die Wahrscheinlichkeit $p(O|W)$ zu bestimmen und dann die wahrscheinlichste Wortfolge auszuwählen (vgl. Abschnitt 2.3).

Zur Berechnung der wahrscheinlichsten Wortfolge kann der *Viterbi-Algorithmus* verwendet werden. Er liefert als Ergebnis diejenige Zustandsfolge zum Modell λ, für die die Wahrscheinlichkeit der Observation maximal ist. Eine Zustandsfolge wird auch *Viterbi-Pfad* genannt, wenn sie durch den Viterbi-Algorithmus berechnet wurde. Die wahrscheinlichste Wortfolge kann gefunden werden, indem untersucht wird, zu welchen Worten die Zustände des Viterbi-Pfades gehören. Dabei lassen sich auch die Wortgrenzen bestimmen, das heißt, die Zeitpunkte zu denen der Viterbi-Pfad von einem Wort in ein anderes wechselt.

Wenn die Äußerung bekannt ist, kann ein spezielles HMM nur für diese Äußerung erzeugt werden, mit dem eine Segmentierung durchgeführt werden kann. Dies wird als *Forced–Alignment* bezeichnet. Basierend auf diesem Verfahren können Spracherkenner trainiert (Viterbi-Training) oder Wort- und Phonemlängen gemessen werden.

Für die Bestimmung des global besten Viterbi-Pfades müssen häufig sehr viele Wahrscheinlichkeiten multipliziert werden. Aus technischen Gründen, Effizienz und Gründen des Wertebereichs werden die Wahrscheinlichkeiten üblicherweise logarithmiert und negiert. Eine so umgeformte Wahrscheinlichkeit wird dann eine *Score* genannt. Die Multiplikationen der Wahrscheinlichkeiten werden durch das Logarithmieren zur Addition. Die umgerechnete Wahrscheinlichkeit einer Zustandsfolge wird als *akkumulierte Score* bezeichnet. Geht die Zustandsfolge vom Wortanfang bis zum Wortende, wird die akkumulierte Score auch *Wort-Score* genannt.

Die Integration eines Sprachmodells erfolgt, indem die Scores für Wortübergänge aus dem Sprachmodell mit einem *Sprachmodellgewicht (LG)* multipliziert werden. Damit wird der Einfluß des Sprachmodells gegenüber dem akustischen Modell gesteuert. Zusätzlich gibt es den Parameter *Wortübergangsstrafterm (WP)*, der auf die Score addiert wird. Mit dem Wortübergangsstrafterm wird gesteuert, ob viele oder wenige Worte in einer Hypothese sein sollen. Das Bestimmen der beiden Parameter erfolgt empirisch und ihre Einstellung hat großen Einfluß auf die Leistungsfähigkeit eines Spracherkenners.

Da die Decodierung eine sehr rechenintensive Aufgabe ist, wurden verschiedene Techniken entwickelt den Suchraum zu beschneiden (*Prunen*). Eine Technik ist die *Strahlsuche* (Beam Search), bei der Zustände aktiv oder inaktiv sein können. Ein Zustand wechselt von aktiv nach inaktiv wenn ein Viterbi-Pfad, der aktuell dort endet, eine sehr viel geringere Wahrscheinlichkeit als der aktuell beste Viterbi-Pfad besitzt. Technisch realisiert wird dies indem, die Differenz der Viterbi-Score des Zustandes zur besten Viterbi-Score gebildet und dieser Wert mit einer Schwelle verglichen wird. Bei der Beschneidung des Suchraumes kann es vorkommen, daß der Viterbi-Pfad beendet wird, der am Ende der Suche Teil des global besten Pfades gewesen wäre. Erkennungsfehler, die darauf beruhen, werden "*Suchfehler*" genannt [16].

2.7 Das Sprachmodell

Die Worte, die ein Spracherkenner erkennen kann, bilden sein Vokabular und sind in einem *Wörterbuch* zusammengefaßt. Dieses Wörterbuch besitzt zusätzlich die Beschreibung des Aufbaus der Worte aus Phonemen. Dieses Wörterbuch kann auch sogenannte *Müllworte* zur Modellierung spontansprachlicher Effekte wie Stottern, Stammeln, Atmen enthalten. Um den Spracherkenner zu unterstützen, wird ein Modell für die erwarteten Äußerungen (Sprachmodell) verwendet.

Ein Typ von Sprachmodell schreibt genau vor, welche Äußerungen möglich sind, indem als Grammatik ein *endlicher Automat* (reguläre Sprache) [23] verwendet wird. Ein solches Sprachmodell schränkt die Verbindungsstruktur der *Wort-*

modelle eines HMM ein. Dieser Ansatz ist für spontane Sprache aber wenig geeignet, da fast jede Wortkombination möglich sein kann, obwohl sie eventuell sehr unwahrscheinlich ist. Ein statistisches Sprachmodell besitzt diese Einschränkung nicht. Es liefert die Wahrscheinlichkeit, daß ein Wort w auftritt basierend auf seiner Historie, also $p(w_m|w_{m-1}w_{m-2}\ldots w_1)$. Diese Wahrscheinlichkeit kann in der Bayes-Formel (2.1) für die Berechnung von $p(W)$ verwendet werden. Dazu wird für die Wortfolge $W = w_1w_2\ldots w_n$ die a priori Wahrscheinlichkeit $p(W)$ durch $\prod_{m=1}^{n} p(w_m|w_{m-1}\ldots w_1)$ berechnet.

Wie gelangt man nun zu einem statistischen Sprachmodell? Gehen wir davon aus, daß es im Deutschen ungefähr 300 000 Worte gibt und die meisten davon sehr selten auftreten. Für die Schätzung der Wahrscheinlichkeiten wird daher eine riesige Menge an Wortfolgen benötigt. Selbst wenn nur eine beschränkte Historie von zwei Worten (Trigramm-Sprachmodell) oder einem Wort (Bigramm-Sprachmodell) berücksichtigt wird, sind die benötigten Datenmengen für eine sichere Schätzung enorm groß. Aus diesem Grund wurden verschiedene Verfahren entwickelt ungesehene Tri- oder Bigramme zu schätzen, indem beispielsweise eine *interpolierte Wahrscheinlichkeit* verwendet, oder eine sogenannte *back-off Wahrscheinlichkeit* [11][33] eingesetzt wird. Ein statistisches Sprachmodell kann dann als Übergangswahrscheinlichkeit in einem HMM von einem Wortmodell zu einem anderen Wortmodell (Bigramm) interpretiert werden.

Wie schwer Sprachmodellierung an sich ist, sei an folgendem Beispiel illustriert. Im Louvre ist der Satz «Hier sehen Sie die Mona Lisa von Leonardo da Vinci» in der Nähe des Bildes sehr viel wahrscheinlicher als auf dem Fischmarkt in Hamburg. Der Satz «Wir schließen in einer Stunde» dagegen ist von der Zeit abhängig und daher an beiden Orten eine Stunde vor dem Schließen wahrscheinlicher als zwei Stunden davor. Mögliche Äußerungen sind von so vielen Faktoren abhängig und besitzen eine so große Vielfalt, daß Sprache nur schwer in Modelle zu fassen ist.

2.8 Training der akustischen Modelle

Das akustische Training dient dazu, die akustischen HMM-Modelle (Worte, Phoneme, Subtriphone) so anzupassen, daß sie einerseits die vorhandene (akustische) Trainingsmenge von Äußerungen gut repräsentieren und andererseits auf ungesehene Äußerungen generalisieren können. Für jedes Modell, das trainiert werden soll, muß eine ausreichende Menge an Trainingsbeispielen vorhanden sein. Üblicherweise ist der erste Schritt das Segmentieren der Trainingsdaten und die Zuordnung welche Modelle damit initialisiert werden sollen. Dieser Arbeitschritt wird *labeln* genannt. Für das *Labeln* gibt es verschiedene Verfahren. Die einfachste Methode ist die Annahme, daß die Zuordnung der Eingabe linear mit der Zahl der Zustände zusammenhängt, die sich aus der Transkription ergeben. Wesentlich besser ist es, die Daten von einem Menschen *labeln* zu lassen. Von Nachteil ist dabei, daß je nachdem wie fein die Segmentierung sein soll, beispielsweise Worte oder Phoneme, nur gut ausgebildete Experten die Segmentierung durchführen können. Zusätzlich ist bei einer feineren Segmentierung der Arbeitsaufwand größer, was dazu führt, daß verhältnismäßig wenig Daten bearbeitet werden können. Ein anderes Problem ist, daß die Segmentierung durch einen Menschen für eine *maschinelle Klassifikation* nicht optimal zu sein braucht. Wegen des großen Aufwands ist es sehr teuer die Daten von Menschen labeln zu lassen. Ist bereits ein (einfaches) Spracherkennungssystem vorhanden, kann dieses verwendet werden, um initiale Labels zu erzeugen. Hierzu wird lediglich eine Transkription benötigt. Da wesentlich mehr Daten verarbeitet werden können und somit mehr Daten für die Initialisierung der Modelle zur Verfügung stehen, ist dies ein bedeutender Vorteil.

Nachdem nun die vorverarbeiteten Sprachdaten segmentiert vorliegen, wird eine *Ballungsanalyse* der Daten für jedes Modell durchgeführt, und für die gefundenen Ballungen eine repräsentative Darstellung erzeugt. Das Ergebnis ist ein initiales *Kodebuch* und eine initiale *Gewichtung*. In einem kontinuierlichen oder semikontinuierlichen HMM ist dies eine parametrische Darstellung aus Mittelwert und Varianz der Ballung (*Kodebuch*) und der a priori Wahrscheinlichkeit der Ballungen in einem gegebenen Zustand des Modells (*Gewichtung*). Beides zusammen nennen wir das *akustische Modell* (λ) des Spracherkenners. In einem semikontinuierlichen HMM besitzen verschiedene Modellzustände gemeinsame Einträge im *Kodebuch*, bei denen sich aber die a priori Verteilungen (*Gewichtung*) unterscheiden.

Besitzen nun alle Modelle ihre initialen *Kodebücher* und *Gewichtungen*, können die Modelle weiter angepaßt werden. Dazu existieren verschiedene Trainingsalgorithmen. Einer ist der *Baum–Welch-Algorithmus* [22][33], der das Optimierungsproblem aus Abschnitt 2.5 löst. Dieser Algorithmus ändert die Modellparameter so ab, daß die modellbedingte Wahrscheinlichkeit $p(O|\lambda)$ aller im Training verwendeten Daten maximiert wird. Es ist nicht garantiert, daß ein globales Maximum gefunden wird. Der Rechenaufwand des *Baum-Welch-Algorithmus* ist im Vergleich zum *Viterbi-Training* groß, da als Ergebnis eine Wahrscheinlichkeit berechnet wird, die die Summe der Wahrscheinlichkeiten aller Pfade ist. Beim Viterbi-Training hingegen wird nur die Wahrscheinlichkeit des besten Pfades berechnet. Dies geschieht indem beim Zusammentreffen mehrerer Pfade nicht die Summe, sondern das Maximum gebildet wird. Wird der wahrscheinlichste Pfad zurückverfolgt, ergibt sich eine Zuordnung einzelner Sprachdatenvektoren zu bestimmten Zuständen. Das Ergebnis ist eine Segmentierung der Daten, mit der erneut eine Ballungsanalyse durchgeführt werden kann.

Mit den gefundenen Verteilungen (Kodebücher und Gewichtung) wird im Dekoder des Spracherkenners die klassenbedingte Wahrscheinlichkeit einer Observation zu einem bestimmten Modellzustand berechnet. Die resultierende Wahrscheinlichkeit kann als Maß für die akustische Ähnlichkeit der Observation zu einem Modell angesehen werden.

2.9 Der Worthypothesengraph

Das Janus-3-Programm verfügt über eine kompakte und praktische Darstellung des Suchraumes, den *Worthypothesengraphen* (Lattice). Im Janus-3-Programm wird eine Lattice aus den Zwischenergebnissen des Decoders (Viterbi-Algorithmus) erzeugt. Ein Worthypothesengraph repräsentiert die Menge der möglichen Hypothesen, die nach Beschneiden des Suchraumes verblieben sind. Eine Besonderheit bei einer Lattice des Janus-3-Programms ist, daß der Einfluß eines Sprachmodells nachträglich variiert oder nachträglich ein anderes Sprachmodell verwendet werden kann.

Ein Worthypothesengraph L besteht aus Knoten V (Vertices) und Kanten E (Edges). Ein Knoten v repräsentiert sowohl ein Wort ($v.Wort$) als auch einen Zeitpunkt ($v.Start$). Eine Kante verbindet jeweils zwei Knoten v_i und v_j *genau dann*, wenn im Decoder ein (partieller) Viterbi-Pfad im Wort $v_i.Wort$ zum Zeitpunkt $v_i.Start$ beginnt und zum Zeitpunkt $v_j.Start - 1$ endet. Eine Kante $e_{i,j}$ besitzt als Attribut die akkumulierte Score (lscore) des Wortes $v_i.Wort$, die im Zeitbereich $v_i.Start$ bis $v_j.Start - 1$ angefallen ist. Dabei handelt es sich *nur* um die akustische Score, da der Anteil des Sprachmodells und der Wortübergangsstrafterm entfernt werden.

Anhand eines Beispiels soll gezeigt werden, wie ein Worthypothesengraph im Janus-3-Programm erzeugt wird. Abbildung 2.6 zeigt eine Darstellung verschiedener Viterbi-Pfade, wie sie beispielsweise auch von Rabiner und Levison [23] oder Ney [15][17] verwendet wird. Von besonderem Interesse ist der Zeitpunkt t_3, da hier zwei verschiedene Pfade gleichzeitig ein Wortende erreichen. Nur ein Pfad wird weiterverfolgt, der andere endet hier. Wird die Darstellung reduziert, so daß nur noch zu erkennen ist, welche Pfade welche Zeitpunkte und Worte miteinander verbinden, ergibt sich Abbildung 2.7. Da üblicherweise die Pfade rückwärts (bezüglich der Zeit) miteinander verbunden sind, wird diese Darstellung auch *Backpointermatrix* genannt. In der Darstellung als Backpointermatrix sind beim Janus-3-Programm die akkumulierten Pfadscores[3] in Knoten (w_i) gespeichert, die Wortenden repräsentieren.

Beim Überführen der Backpointermatrix in eine Lattice werden nun Knoten (v_i) verwendet, die Wortanfänge repräsentieren; Kanten speichern nun die lscores.

[3] die ebenfalls die Scores des Sprachmodells beinhalten

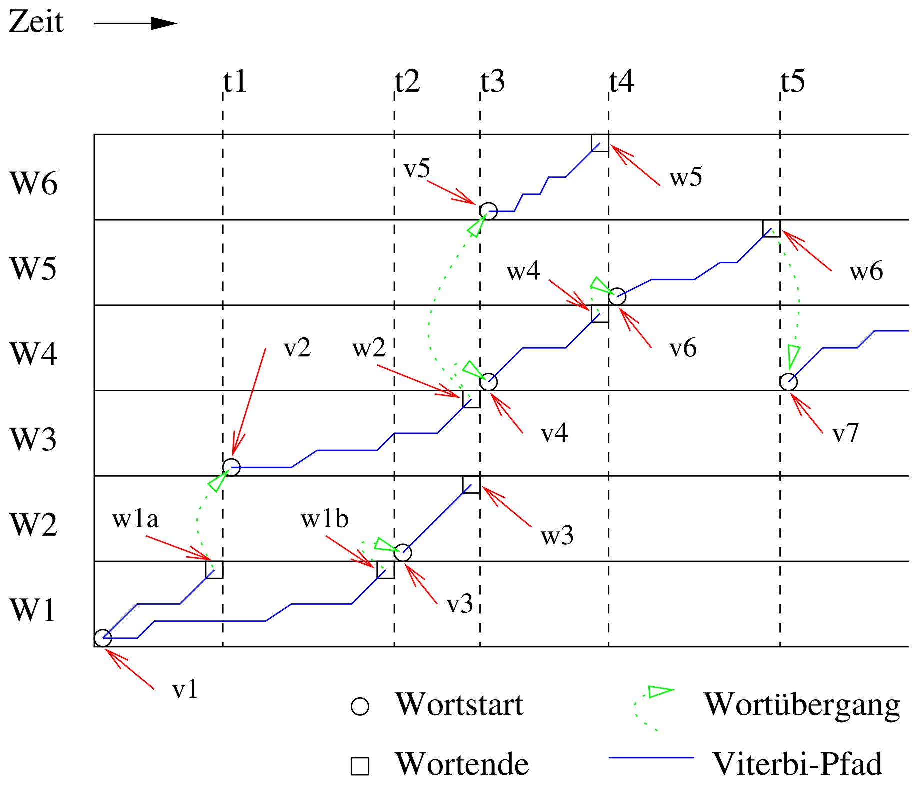


Abbildung 2.6: Suchraum mit Viterbi-Pfaden

Die lscore wird dabei als Differenz der Pfadscores der Wortende-Knoten berechnet, und die Score des Sprachmodells subtrahiert. Beim Erzeugen der Kanten werden auch weniger wahrscheinliche Hypothesen in die Lattice aufgenommen, da *alle* Pfade, die ihr Wortende erreichten, mit *allen* nachfolgend beginnenden Worten verbunden werden. Dadurch werden auch Verbindungen eingefügt, die in der Backpointermatrix nicht bestanden, beispielsweise (v_3, v_4) und (v_5, v_6). In der Backpointermatrix entspräche dies dem Einfügen von Kanten zwischen (w_4, w_3) und (w_6, w_5). Die Kanten (v_2, v_4) und (v_2, v_5) besitzen die gleiche lscore, da beide den gleichen Viterbi-Pfad darstellen. Als zusätzliche Knoten besitzen sowohl die Lattice als auch die Backpointer-Matrix besondere Knoten, die den Start v_a und das Ende v_e einer Äußerung darstellen.

Zur Verdeutlichung der Besonderheiten der Lattice des Janus-3-Programms sind Teile der Datenstruktur in Tabelle 2.1 und Tabelle 2.2 angegeben. Die in Klammern angegebenen Namen sollen das Auffinden im Quelltext erleichtern. Um einen schnellen Zugriff auf Kanten zu haben, die von zeitlich früher liegenden Knoten kommen, wurde die Datenstruktur der Lattice für diese Arbeit erweitert (rlinkP, rnextP). Eine schematische Darstellung der Zugriffsmöglichkeiten zwischen den Elementen einer Lattice zeigt Abbildung 2.9.

Die Lattice kann zur Berechnung einer Hypothese mit geänderten Sprachmo-

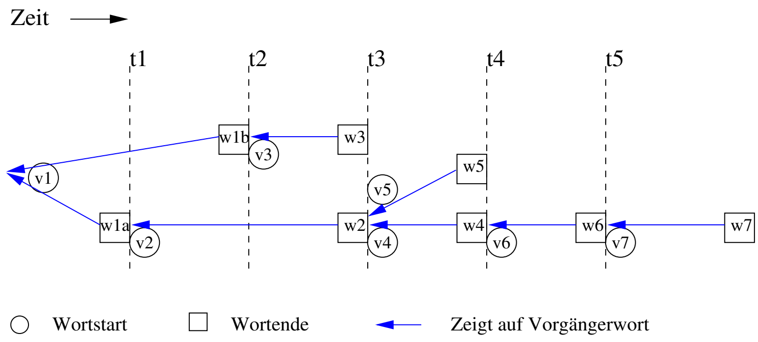


Abbildung 2.7: Darstellung der Viterbi-Pfade als Graph

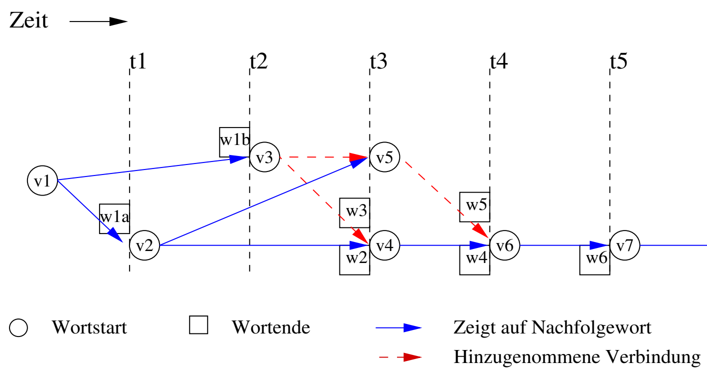


Abbildung 2.8: Die resultierende Lattice

dellparametern (LG,WP) oder geändertem Sprachmodell verwendet werden. Da das die Lattice erzeugende Sprachmodell[4] entfernt wurde, wird hierfür wieder ein Sprachmodell benötigt. Der Einfluß des Sprachmodells, das die Lattice erzeugt läßt sich allerdings rechnerisch nicht völlig entfernen. Es ist mitverantwortlich welche Wortübergänge stattfinden und welche Viterbi-Pfade aus dem Suchraum entfernt werden. Seine Strukturen werden in die Lattice eingeprägt.

Zur Berechnung der neuen Hypothese wird der wahrscheinlichste (ko-

[4] d.h. das im Decoder verwendete

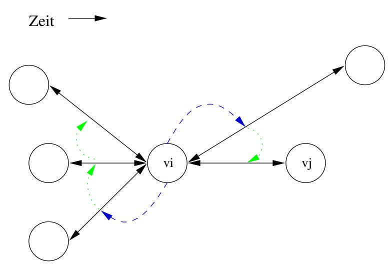


Abbildung 2.9: Verbindungsstruktur der Lattice–Elemente

stengünstigste) Pfad durch den Worthypothesengraphen gesucht. Dies ist ein Single-Source-Shortest-Path-Problem in einem azyklischen Graphen [14][3][19], als Kosten treten die lscores und Scores aus dem Sprachmodell auf. Die Berechnung erfolgt analog dem Viterbi-Algorithmus, wobei die akkumulierten Pfadscores (pscore) in den Kanten gespeichert werden. Der so gefundene beste Pfad ist dann die Hypothese des Spracherkenners.

Wort	kennzeichnet welches Wort der Knoten repräsentiert (Wortindex)
Start	Zeitpunkt an dem der Viterbi-Pfad das Wortmodell betreten hat (frameX)
AusKantenListe	Verweis auf die Liste der Kanten, die den Knoten mit zeitlich folgenden Worten verbindet (linkP)
InKantenListe	Verweis auf den Start der Liste der Kanten, die den Knoten mit zeitlich vorhergehenden Knoten verbindet (rlinkP)

Tabelle 2.1: Attribute eines Knotens

2.10 Entropie

Angenommen es gibt eine Menge möglicher Ereignisse, beispielsweise Wortausprägungen, oder eine Menge von Zuständen in einem *HMM*, die mit den Wahrscheinlichkeiten $p_1, p_2, ...p_n$ auftreten können. Ist nun *unbekannt* welches Ereignis konkret auftreten wird oder aufgetreten ist, so kann gefragt werden, wie groß die Unsicherheit ist, welches der möglichen Ereignisse eintreten wird oder eingetreten ist. Ein Maß, das diese Unsicherheit bewertet, stammt aus der *Informationstheorie* und wurde von Shannon [30] genauer untersucht. Folgende Anforderungen werden von ihm an ein solches Maß H gestellt:

1. H soll stetig von den p_i abhängen.
2. Sind die p_i gleichverteilt, also $p_i = \frac{1}{n}$, soll H monoton mit n wachsen.
3. Wenn ein Ereignis in zwei aufeinanderfolgende Ereignisse zerlegt werden kann soll sich das ursprüngliche H aus einer gewichteten Summe der Unsicherheiten jeder einzelnen Entscheidung ergeben. Abbildung 2.10 zeigt eine

[5] Das optionale Stillemodell soll kurze Pausen zwischen Worten modellieren. In kontinuierlicher Sprache treten Pausen zwischen Worten aber nicht immer, sondern nur optional auf.

VonKnoten	Startknoten der Kante (zeitlich früher) (fromP)
ZuKnoten	Zielknoten der Kante (zeitlich später) (toP)
WortEnde	Zeitpunkt an dem das Wort, das beim Startknoten begonnen hat, endet. Dieser Wert braucht *nicht* unbedingt der Zeitpunkt direkt vor Beginn des nächsten Wortes (ZuKnoten.Start) zu sein, sondern kann eine größere Lücke aufweisen. Die Lücke tritt dann auf, wenn der Viterbi-Pfad am Ende eines Wortes das optionale Stillemodell betreten hat[5]. (frameX)
lscore	lokale akustische Score (lscore)
pscore	enthält die akkumulierte Pfadscore, die für die Bestimmung einer Hypothese benötigt wird. Diese Score beinhaltet auch einen Anteil aus einem Sprachmodell. (pscore)
NächsteInKante	Verweis auf die nächste Kante in der Liste der Kanten mit gemeinsamem *VonKnoten* (nextP)
NächsteAusKante	Verweis auf die nächste Kante in der Liste der Kanten mit gemeinsamem *ZuKnoten* (rnextP)

Tabelle 2.2: Attribute einer Kante

solche Aufteilung eines Ereignisses. Die Beziehung, die nun gelten soll, ist $H(\frac{1}{2},\frac{1}{3},\frac{1}{6}) = H(\frac{1}{2},\frac{1}{2}) + \frac{1}{2}H(\frac{2}{3},\frac{1}{3})$. Dabei ergibt sich der Gewichtungsfaktor $\frac{1}{2}$, weil diese zweite Wahl nur halb so oft getroffen wird.

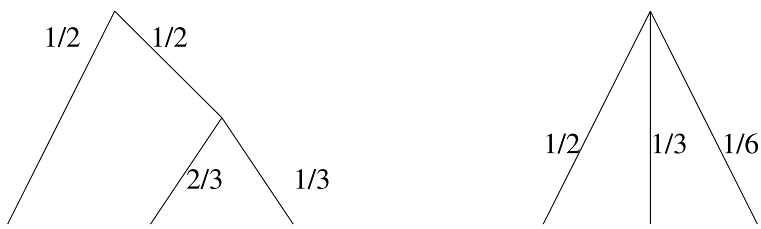


Abbildung 2.10: Umwandlung einer Mehrfachentscheidung

In [30] ist folgender Satz bewiesen:

Satz 1 *Der einzige Wert H, der die oben geforderten Voraussetzungen erfüllt, hat die Form*

$$H = -K \sum_{i=1}^{n} p_i \log(p_i) \tag{2.4}$$

dabei ist K eine positive Konstante.

Formel 2.4 spielt in der *Informationstheorie* eine bedeutende Rolle. Dort wird der Logarithmus zur Basis zwei und K als eins gewählt. Die Einheit des Ergebnisses wird *Bit* genannt und stellt ein Maß für den Informationsgehalt der Quelle dar. Der Wert, der sich dann aus der Formel 2.4 ergibt, wird als *Entropie des Wahrscheinlichkeitssatzes* $p_1, p_2, ..., p_n$ bezeichnet.

Aus der *Informationstheorie* [30] ist bekannt, daß es dann eine Kodierung gibt, die im *Mittel* bis auf ein beliebig kleines $\epsilon > 0$ *H Bit je Zeichen* benötigt. Das bedeutet je weniger *Bit* benötigt werden, desto weniger Auswahlmöglichkeiten gibt es im Mittel, was eine Interpretation als geringe Unsicherheit ermöglicht. Sind beispielsweise 16 Ereignisse möglich, die wir aber im Mittel mit nur einem *Bit* kodieren können, so ist die Unsicherheit über das nächste Ereignis genauso groß wie wenn nur zwei Ereignisse möglich wären, diese aber gleichwahrscheinlich sind.

Weitere Eigenschaften von H sind [30]:

1. Es gilt $H = 0$ genau dann, wenn alle p_i bis auf eines 0 sind.
2. Für ein gegebenes n ist H maximal und gleich $\log n$, wenn für alle i gilt $p_i = \frac{1}{n}$.
3. Jede Veränderung in Richtung Gleichverteilung der Wahrscheinlichkeiten $p_1, p_2, ..., p_n$ läßt H wachsen.

Zur letzten Eigenschaft sei noch folgende Rechnung angeführt, bei der ein Ereignis eine hohe Wahrscheinlichkeit besitzt, beispielsweise $\frac{1}{2}$, und die restliche Wahrscheinlichkeit sich auf $N \geq 1$ gleichwahrscheinliche Ereignisse verteilt, also jedes Ereignis mit einer Wahrscheinlichkeit von $\frac{1}{2N}$ eintreten kann. Die Entropie beträgt hierfür

$$\begin{aligned} & -(\frac{1}{2}\log\frac{1}{2} + \sum_{1}^{N}\frac{1}{2N}\log\frac{1}{2N}) \\ = & -(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2N}) \\ = & \ 1 + \frac{1}{2}\log N \end{aligned}$$

Obwohl ein mögliches Ereignis stark dominiert, kann die Unsicherheit über das aufgetretene Ereignis beliebig wachsen, wenn N beliebig groß werden kann. Durch das Beschneiden des Suchraumes im Decoder eines Spracherkenners (Abschnitt 2.6) tritt der umgekehrte Effekt ein. Dabei kann eine beliebig große Unsicherheit klein werden.

Wird in Formel 2.4 eine Verbundwahrscheinlichkeit $p(x, y)$ eingesetzt, ergibt sich die Verbundentropie $H(X; Y)$. Wird dagegen eine bedingte Wahrscheinlichkeit $p(y|x)$ eingesetzt, ergibt sich die bedingte Entropie $H(Y|X)$. Die bedingte Entropie ist ein Maß dafür "wie sicher wir im Durchschnitt über y sind, wenn wir x kennen"[30] und kann für die Berechnung der Entropie in einem Graphen eingesetzt werden.

In 2.11 ist ein gerichteter Graph ohne Zyklen dargestellt, dessen Kanten mit Übergangswahrscheinlichkeiten für Ereignisse attribuiert sind. Die *Entropie* ist nun ein Maß für die Unsicherheit, gemessen über alle möglichen Pfade des Graphen. Ein äquivalenter Baum bezüglich der möglichen Ereignisfolgen und deren zugehörigen Wahrscheinlichkeiten ergibt sich durch die Duplizierung gemeinsamer Teilgraphen. Das Ergebnis der Umformung des Graphen aus Abbildung 2.11 ist in Abbildung 2.12 zu sehen. Um die Berechnung mit Formel 2.4 durchführen zu können, kann dieser Baum weiter umgeformt werden, so daß sich als Ergebnis der Baum aus Abbildung 2.13 ergibt. Hierzu wurden lediglich die aus der Wahrscheinlichkeitsrechnung bekannten Regeln der bedingten Wahrscheinlichkeiten angewandt.

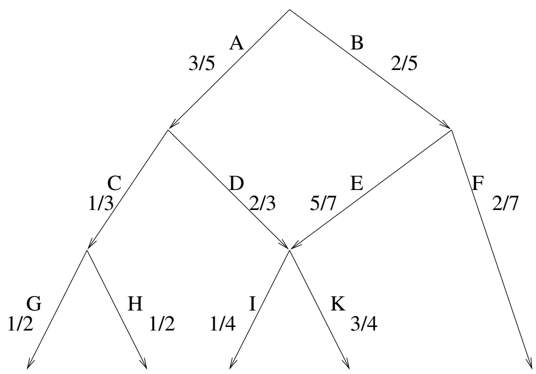


Abbildung 2.11: Gerichteter Graph mit Übergangswahrscheinlichkeiten

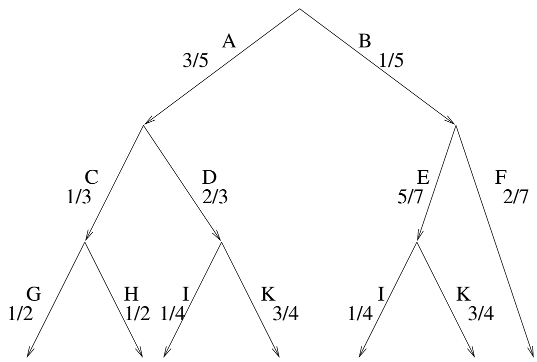


Abbildung 2.12: Baum mit Übergangswahrscheinlichkeiten

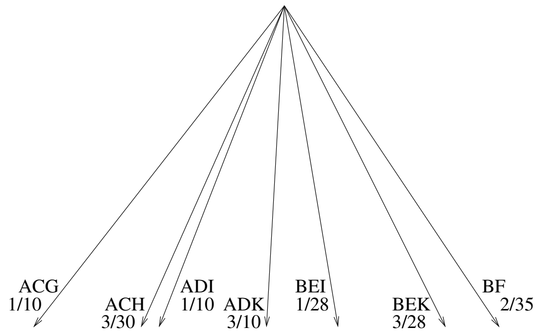


Abbildung 2.13: Vereinfachter Baum mit Übergangswahrscheinlichkeiten

Eine Berechnung nach dem eben beschriebenen Verfahren empfiehlt sich jedoch nicht, da bereits bei einfachen Graphen die Zahl der möglichen Pfade stark anwachsen kann. Abbildung 2.14 zeigt ein Beispiel für einen Graphen, der diese Eigenschaft besitzt. Hier hängt die Zahl der Kanten nur linear von der Zahl der Knoten ab. Die Zahl der Pfade zu einer gegebenen Knotenanzahl n läßt sich in diesem Fall mit folgender Formel berechnen:

$$F(0) = 1; F(1) = 1; F(n) = F(n-1) + F(n-2) \text{ für } n \geq 2 \tag{2.5}$$

Diese Formel berechnet die berühmten Fibonacci-Zahlen, von denen bekannt ist, daß sie exponentiell wachsen.

In der Spracherkennung können Graphen wie in Abbildung 2.14 als Teilgraphen in Worthypothesengraphen auftreten. Bei den akustischen Modellen ist es ebenfalls üblich, HMM mit ähnlicher Struktur für Phoneme zu verwenden.

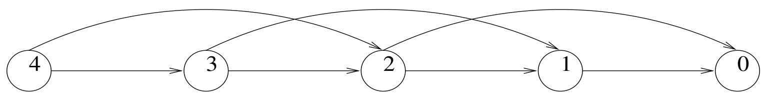


Abbildung 2.14: Struktur mit exponentieller Anzahl von Pfaden

Für eine effiziente Berechnung ist die oben geforderte Beziehung bezüglich mehrfacher Entscheidungen auszunutzen und H rekursiv zu berechnen. Dies allein führt jedoch noch nicht zu einer Beschleunigung, wenn nicht zugleich die Methode des *dynamischen Programmierens* angewendet wird. Eine mehrfache Berechnung von Teilergebnissen wird dabei vermieden. Hat ein Graph viele Knoten an denen Pfade zusammenfließen, dann muß die Entropie H für die darunterliegenden Teilgraphen nur einmal berechnet werden. In Abbildung 2.11 gilt dies beispielsweise für den Knoten, bei dem die Kanten D und E zusammentreffen. Der Aufwand hängt nun im wesentlichen von der Anzahl der Kanten im Graphen ab.

Kapitel 3

Bewertungsmöglichkeiten für Vertrauensmesser

In diesem Kapitel werden Maße für die Leistungsfähigkeit von Vertrauensmessern behandelt. Zunächst wird erklärt wie festgestellt wird, ob ein Erkennungsfehler vorliegt und wie sich die Fehler einteilen lassen. Anschließend werden die allgemeinen Freiheitsgrade und Schwierigkeiten erörtert, die bei der Gestaltung eines Vertrauensmessers für die maschinelle Spracherkennung gegeben sind. Abschließend werden verschiedene Maße zur Bewertung der Leistung eines Vertrauensmessers diskutiert.

3.1 Der Align-Algorithmus

Bei der Spracherkennung gibt es Situationen, in denen eine Zuordnung der Worte zwischen der tatsächlich gesagten Äußerung (Referenzsatz) und dem vom Spracherkenner erzeugten Hypothesensatz benötigt wird. Der Editierabstand ist die *minimale* Anzahl der Operationen *Wort-Einfügen*, *Wort-Löschen* und *Wort-Ersetzen*, die nötig ist, um den Referenzsatz in den Hypothesensatz zu überführen. Zur Bestimmung dieser Operationen wird der Align-Algorithmus eingesetzt. Anhand der Operationen ist es möglich, Erkennungsfehler weiter zu unterscheiden.[1] Ein Wort, das in den Referenzsatz eingefügt wird, gehört zur Klasse der *Einfügefehler*, eines, das aus dem Referenzsatz gelöscht wird, zur Klasse der *Löschfehler*. Einfügefehler treten zwischen Worten des Referenzsatzes, Löschfehler zwischen den Worten der Hypothese auf. Ein Wortpaar aus Referenz- und Hypothesensatz, das durch die Operation Wort-Ersetzen gebildet wird, gehört dementsprechend zur Klasse der *Ersetzungsfehler*.

[1] Aufgrund der Minimalität können sich die Operationen nicht aufheben.

Sei beispielsweise der Referenzsatz «Das hier ist der erste Satz» und der Hypothesensatz «Das ist aber der zweite Satz», so gibt es mehrere Möglichkeiten wie die minimale Anzahl von drei Operationen erreicht werden kann (vgl. Tabelle 3.1 und 3.2).

Referenz	Hypothese	Fehlertyp
Das	Das	Korrekt
hier	ist	Ersetzung
ist	aber	Ersetzung
der	der	Korrekt
erste	zweite	Ersetzung
Satz	Satz	Korrekt

Tabelle 3.1: Fehlerzuordnung 1

Referenz	Hypothese	Fehlertyp
Das	Das	Korrekt
hier	*	Löschung
ist	ist	Korrekt
*	aber	Einfügung
der	der	Korrekt
erste	zweite	Ersetzung
Satz	Satz	Korrekt

Tabelle 3.2: Fehlerzuordnung 2

Bei der Untersuchung, welche Merkmale auf eine fehlerhafte Erkennung hinweisen können, ist es von großer Bedeutung, wie die Fehler zugeordnet werden. Im angeführten Beispiel werden bei der Zuordnung nach Tabelle 3.2 zwei Worte der Hypothese als falsch angesehen, nach der Zuordnung mit Tabelle 3.1 drei. Welche der Zuordnungen den tatsächlichen (zeitlichen) Gegebenheiten entspricht, kann a priori nicht entschieden werden. Eine Zuordnung jedoch, die möglichst wenig fehlerhafte Worte im Hypothesesatz erzeugt, scheint im allgemeinen die plausiblere zu sein.

Der Align-Algorithmus in obigem Beispiel würde die Lösung aus Tabelle 3.2 erzeugen. Die Operationen Wort-Einfügen und Wort-Löschen werden bevorzugt, da beide mit geringeren Kosten bewertet sind als die Operation Wort-Ersetzen (vgl. Tabelle 3.3).

Operation	Kosten
Wort-Einfügen	75
Wort-Löschen	75
Wort-Ersetzen	100

Tabelle 3.3: Kosten der Operationen bei Editierdistanz

Die Arbeitsweise des Align-Algorithmus ist in Abbildung 3.1 dargestellt, sie entspricht etwa dem Viterbi-Algorithmus. Dabei muß ein Pfad mit minimalen Kosten in der dargestellten Matrix berechnet werden. Treffen mehrere Pfade zusammen wird nur der kostengünstigste weiter verfolgt. Wird der Pfad, der die beiden Satzenden erreicht (Matrix oben, rechts), zurückverfolgt, können die benötigten Operationen (gestrichelten Pfeile) bestimmt werden. In der Abbildung ist nur der günstigste Pfad eingezeichnet.

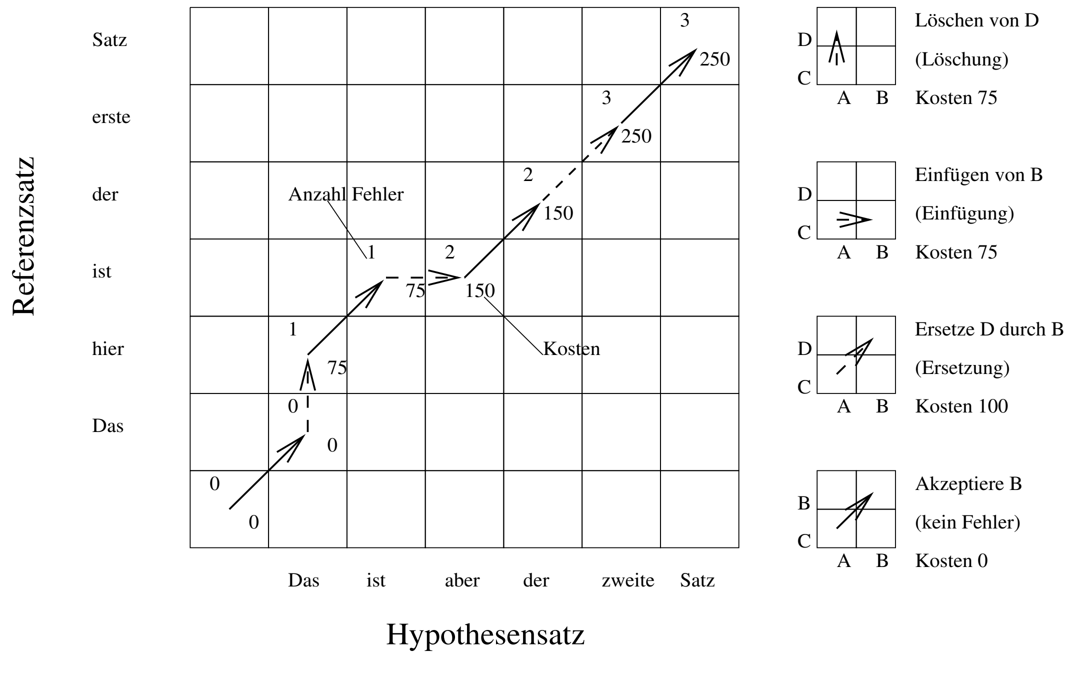


Abbildung 3.1: Arbeitsweise des Align-Algorithmus

Der Editierabstand, der durch den Align-Algorithmus bestimmt wird, ist in der Spracherkennung von großer Bedeutung, da mit ihm ein anerkanntes Maß zur Beurteilung der Qualität eines Spracherkenners berechnet werden kann.

Die Wort-Akkuratheit (WA) eines Spracherkenners ist definiert als

$$\text{WA} = \frac{\#Ersetzungen + \#Einfügungen + \#Löschungen}{\#Worte\_im\_Referenzsatz} \tag{3.1}$$

Eine Unterscheidung fehlerhafter Worte des Hypothesensatzes in Einfüge- und Ersetzungsfehler erscheint für eine Klassifikation durchaus sinnvoll, ist aber mit dem hier dargestellten Align-Algorithmus nicht sicher durchführbar. Treten beispielsweise Einfüge- und Ersetzungsfehler nebeneinander in einer Hypothese auf, ist die Reihenfolge (Einfügung/Ersetzung, Ersetzung/Einfügung) dem Algorithmus gleich teuer. Insbesondere können diese Kombinationen dann auftreten, wenn ein langes Wort in mehrere ähnlich klingende, kurze Worte zerteilt wird. Wird beispielsweise das Wort «füreinander» zu «für ein anderes» sind dies zwei Einfügungen und eine Ersetzung. Solche Effekte treten insbesondere dann auf, wenn das zu erkennende Wort des Referenzsatzes nicht im Vokabular des Erkenners vorhanden ist. Dazu gehören häufig Namen oder beispielsweise auch das Wort «füreinander» im verwendeten Spracherkenner. Ersetzungsfehler lassen sich somit nochmals unterteilen in *normale Ersetzungfehler* und sogenannte *OOV-Fehler* (OOV = Out of Vocabulary). Das Beispiel zeigt aber, daß Einfügefehler (im Prinzip) ebenfalls OOV-Fehler sein können.

Abschließend sei bemerkt, daß *vor* dem Durchführen der Zuordnung alle Müllworte (#noise#, #atmen#) aus der Hypothese und der Transkription entfernt wurden. Dies ist wichtig, da diese häufig nicht in der Transkription enthalten sind und somit die Zahl der möglichen Zuordnungen steigt, denn eingefügte Müllworte werden nicht automatisch als Einfügung erkannt. Es kann vorkommen, daß Worte, die durch viele Einfügungen von Müllworten eingekreist sind, unnötigerweise als Ersetzungsfehler angesehen werden, wodurch die Zahl der fehlerhaften Worte in der Hypothese steigt.

3.2 Einteilung von Vertrauensmessern

Bevor Bewertungsmaßstäbe beschrieben werden, mit denen die Leistungsfähigkeit von Vertrauensmessern beurteilt wird, sollen diese Vertrauensmesser noch einmal weiter unterteilt werden. Als Kriterium der Einteilung werden zum einen die möglichen Ausgabewerte eines Vertrauensmessers herangezogen, zum anderen, ob die zeitliche Zuordnung berücksichtigt wird.

Die Ausgabe eines Vertrauensmesser kann beispielsweise binär sein, das bedeutet, falls ein Wort eher als falsch angesehen werden soll, produziert der Vertrauensmesser zu diesem Wort die Marke 0 (0 %), sonst die Marke 1 (100 %), Zwischenwerte gibt es nicht. Wir nennen dies einen *binären Vertrauensmesser*.

Ein *kontinuierlicher Vertrauensmesser* produziert als Ausgabe einen "beliebigen" Wert, der im Bereich von 0 % bis 100 % liegt, und als a posteriori Wahrscheinlichkeit für eine korrekte Erkennung des Wortes w interpretiert werden kann ($p(Korrekt|w)$). Ein kontinuierlicher Vertrauensmesser kann durch Verkettung mit einer Schwellwertfunktion zu einem binären Vertrauensmesser werden.

Einen Vertrauensmesser, der die Worte eines Hypothesensatzes ohne deren zeitliche Zuordnung bewertet, bezeichnen wir als einen *Align-Vertrauensmesser*, da hier allein die Zuordnung des Referenzsatz durch den Align-Algorithmus eine Rolle spielt.

Es ist aber auch möglich, strenger als der Align-Algorithmus zu sein und "starke" Abweichungen bei der zeitlichen Übereinstimmung ebenfalls als Fehler oder Teilfehler anzusehen. Wie eine zeitliche Zuordnung des Satzes gefunden werden kann ist in Abschnitt 2.6 beschrieben.

Betrachten wir beispielsweise den Referenzsatz ≪Das is' so ordentlich≫ und eine mögliche Hypothese ≪Das is' ordentlich≫. Nach dem Align-Algorithmus sind alle Worte der Hypothese korrekt erkannt worden, eine Löschung ist aufgetreten. Sei nun aber die zeitlichen Zuordnungen der Worte wie sie in Abbildung 3.2 dargestellt ist. Das Wort ≪so≫ wäre von seinen Nachbarworten "geschluckt" worden und es ist nicht einzusehen, weshalb diese Worte nun immer noch als *völlig* korrekt anzusehen sind. Eine ungelöste Frage ist, wie "falsch" solche Wort/Zeit-Paare nun aber sind.

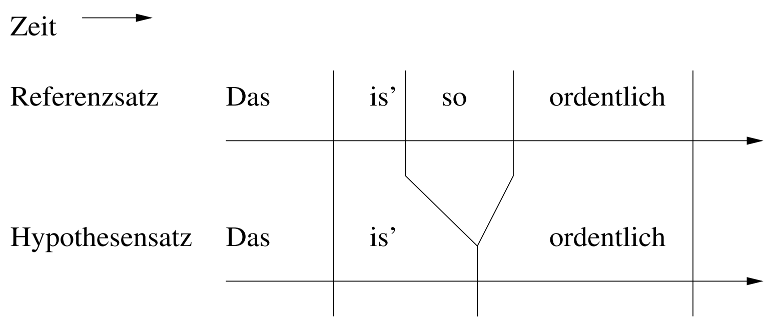


Abbildung 3.2: Zeitliche Zuordnung von Referenz- und Hypothesensatz

Ein einfacher Ansatz ist beispielsweise Löschfehler dieser Art auf die Nachbarworte anteilig der Zeit aufzuteilen und das ebenso mit Einfügefehlern zu machen. Diese Lösung bereitet aber insbesondere dann Probleme, wenn mehrere Lösch- oder Einfügefehler auf ein einzelnes Wort der Hypothese fallen. Eine befriedigende Lösung für das Problem ist nicht offensichtlich. Einen Vertrauensmesser für Wort/Zeit-Paare nennen wir *Zeitalign-Vertrauensmesser*. Der Vorteil eines solchen Vertrauensmesser ist, daß sich Löschfehler leichter zuordnen lassen, da der Zeitraum bekannt ist, in dem ein Löschfehler auftritt.

Aufgrund des ungelösten Problems wenn bei der Fehlerzuordnung die zeitliche Zuordnung berücksichtigt werden soll, wird in dieser Arbeit nur ein Align-Vertrauensmesser betrachtet.

3.3 Qualitätsmaße für Vertrauensmesser

Die in folgender Tabelle zusammengefaßten Definitionen werden in diesem Abschnitt benötigt:

Abkürzung	Bedeutung
k/f	korrekt/falsch
$\mathcal{K}_H$	Menge aller korrekten Worte der Hypothese
$\mathcal{F}_H$	Menge aller falschen Worte der Hypothese
$\mathcal{W}_H$	Menge aller Worte in der Hypothese (ohne Müllworte)
$\mathcal{T}_k$	dito, von einem binären Vertrauensmesser als korrekt markiert
$\mathcal{T}_f$	analog als falsch markiert
N	$\|\mathcal{W}_H\|$ Anzahl der zu markierenden Worte

Tabelle 3.4: Verwendete Notation

Seien unterschiedliche Vertrauensmesser gegeben, die beispielsweise verschiedene Merkmale verwenden. Soll bestimmt werden, welcher der Vertrauensmesser am besten ist, wird ein *objektives* Qualitätsmaß (QM) benötigt, mit dem die verschiedenen Vertrauensmesser verglichen werden können. Anhand eines solchen Qualitätsmaßes kann beispielsweise festgestellt werden, welche der Merkmalskombination für eine Klassifikation am geeignetsten ist. Besonders vorteilhaft ist es, wenn sich dieses Maß durch eine einzige Kennzahl ausdrücken läßt. Das ist aber bei vielen hier betrachteten Qualitätsmaßen nicht der Fall. So ist für einige Qualitätsmaße ein a priori Vertrauenmesser notwendig, um eine Aussage machen zu können, ob überhaupt eine Verbesserung erreicht wurde (Baseline).

Ein anderer Aspekt ist, daß es Qualitätsmaße gibt, die nur für binäre Vertrauensmesser sinnvoll sind. Das bedeutet, daß für einen Vertrauensmesser, der a posteriori Wahrscheinlichkeiten mißt, für die Korrektheit noch eine Schwellwertfunktion $p(k|w)$ nachgesetzt werden muß.

Das binäre Qualitätsmaß

Ein Qualitätsmaß für einen binären Vertrauensmesser berechnet sich, indem die Anzahl der korrekten Marken durch die Gesamtzahl der Marken geteilt wird. Wir bezeichnen es als *binäres Qualitätsmaß* (B-QM).

$$\text{B-QM} = \frac{|(\mathcal{K}_H \cap \mathcal{T}_k) \cup (\mathcal{F}_H \cap \mathcal{T}_f)|}{\mathcal{W}_H} \tag{3.2}$$

Dieses Maß ist deutlich von der Qualität des Spracherkenners abhängig. Es stellt sich nun die Frage, wie gut der beste a priori Klassifikator ist, und welche Ausgabe er haben muß. Da ein a priori Klassifikator hier nur zwei mögliche Ausgaben haben kann (nämlich 0 % für falsch und 100 % für korrekt), ist sofort einzusehen, daß der Wert für die Klasse ausgegeben werden soll, die häufiger ist: 100 %, wenn über 50 % der zu markierenden Worte korrekt sind, und 0 % sonst. Als Baseline für das B-QM erhält man dann den Anteil korrekter bzw. falscher Worte in der Hypothese.[2]

Das kontinuierliche Qualitätsmaß

Eine mögliche Verallgemeinerung auf Klassifikatoren, die a posteriori Wahrscheinlichkeiten schätzen, wäre das arithmetische Mittel der Richtiganteile zu verwenden, um ein *kontinuierliches Qualitätsmaß* (C-QM) zu erhalten.

$$\text{C-QM} = \frac{\sum\limits_{w \in \mathcal{K}_H} p(k \mid w) + \sum\limits_{w \in \mathcal{F}_H} (1 - p(k \mid w))}{|\mathcal{W}_H|} \tag{3.3}$$

Nun stellt sich die Frage, welche Ausgabe ein a priori Klassifikator haben müßte, der C-QM maximiert. Dazu sei angenommen, daß p der Anteil an korrekten Worten sei und t der Ausgabewert, des a priori Klassifikators ($t \in [0, 1]$).

$$\begin{aligned} \text{C-QM}(t) &= \frac{pt + (1-p)(1-t)}{1.0} \\ &= t(2p - 1) + 1 - p \end{aligned} \tag{3.4}$$

C-QM(t) ist also eine lineare Funktion in t. Sei der Anteil korrekter Worte p > 50 %, dann wird das globale Maximum mit t = 100 % erreicht, wenn also der a priori Vertrauensmesser alle Worte als korrekt ansieht.

Daraus folgt für einen Klassifikator, daß C-QM vergrößert werden kann indem 100 % ausgegeben wird wenn die a posteriori Wahrscheinlichkeit für $p(k|w)$ größer als 50 % ist und sonst 0 % ausgegeben wird. Das Ergebnis entspricht dann aber genau dem binären Qualitätsmaß. C-QM ist somit als kontinuierliches Qualitätsmaß ungeeignet.

---

[2]Da der verwendete Spracherkenner über 50 % der Worte in den Hypothesen korrekt hat, sollte der a priori Klassifikator also stets "100 %" ausgeben.

Das geometrische Mittel

Wird statt des arithmetischen Mittels dagegen das geometrische Mittel der Korrektanteile verwendet, erhält man ein Qualitätsmaß, das für kontinuierliche Vertrauensmesser verwendbar ist, wir nennen es GM-QM.

$$\text{GM-QM} = \sqrt[N]{\prod_{w \in \mathcal{K}_H} p(k|w) \prod_{w \in \mathcal{F}_H} (1 - p(k|w))} \tag{3.5}$$

$$= 2^{\frac{1}{N}(\sum_{w \in \mathcal{K}_H} \log_2 p(k|w) + \sum_{w \in \mathcal{F}_H} \log_2(1-p(k|w)))} \tag{3.6}$$

Die umgeformte Darstellung des GM-QM weist einen Zusammenhang zur bedingten Entropie[3] $H(X|Y)$ auf; diese wird für hinreichend großes N durch den Exponenten approximiert.[4]

Dadurch ergibt sich eine Interpretation eines Vertrauensmessers als Teil eines Kommunikationskanals, wie er in Abbildung 3.3 dargestellt ist. Als Eingabe erhält der Kanal eine Folge von 0en und 1en, wobei 0 für ein falsch erkanntes Wort und 1 für ein korrekt erkanntes Wort steht.

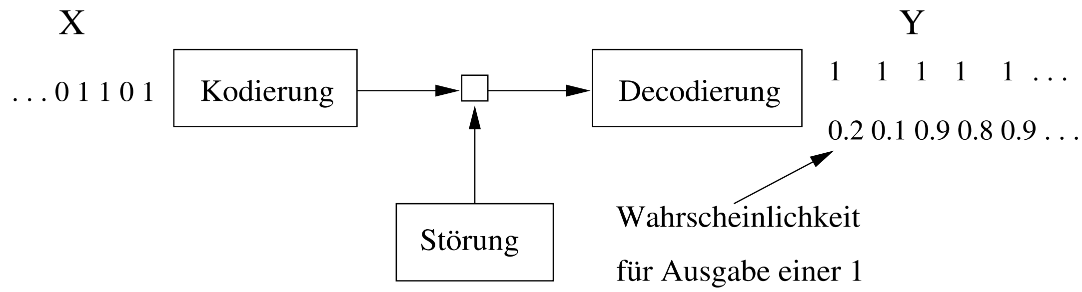


Abbildung 3.3: Kommunikationskanal

Die Hypothese stellt damit eine Kodierung einer Folge von 0en und 1en dar. Der Vertrauensmesser entspricht dem Decoder und ist Teil des Kanals. Die Ausgabe des Decoders in Abbildung 3.3 stellt die korrespondierende Folge der a posteriori Wahrscheinlichkeiten $p(k|w)$ dar, also ob das Wort w der Hypothese einer "gesendeten" 1 entspricht.

Das kontinuierliche Entropie-Qualitätsmaß

Es soll nun der Zusammenhang zwischen der bereits erwähnten bedingten Entropie und dem Exponenten aus Formel 3.6 aufgezeigt werden. Dazu sollen folgende Vereinbarungen gelten: X ist eine Zufallsvariable mit dem Wertebereich

[3] Äquivokation nach Shannon [30]
[4] falls die Quelle ergodisch ist

$\mathcal{X} = \{0, 1\}$, die die Nachrichtenquelle darstellt und die nur die Werte 1 für korrekt (k) und 0 für falsch (f) annimmt. Y ist eine Zufallsvariable mit dem Wertebereich $\mathcal{Y} = \{0, 1\}$, die die Ausgabe des Kanals darstellt. Mit $p(x, y)$ bezeichnen wir die Verbundwahrscheinlichkeit, also daß x und y gemeinsam auftreten. Außerdem existiert noch folgende Hilfsfunktion $c_x = 1$ falls $x = 1$, sonst $c_x = 0$.

$$\begin{aligned}
& H(X|Y) \\
= \quad & \sum_{y\in\mathcal{Y}} p(y) H(X|Y = y) && (3.7) \\
= \quad & -\sum_{y\in\mathcal{Y}} p(y) \sum_{x\in\mathcal{X}} p(x|y) \log_2 p(x|y) && (3.8) \\
= \quad & -\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} p(x, y) \log_2 p(x|y) && (3.9) \\
= \quad & -E_{p(x,y)}(\log p(x|y)) && (3.10) \\
\approx \quad & -\frac{1}{N} \sum_{i=1}^{N} \log_2 p(x_i|y_i) && (3.11) \\
= \quad & -\frac{1}{N} \sum_{i=1}^{N} (c_{x_i} \log_2 p(1|y_i) + (1 - c_{x_i}) \log_2 p(0|y_i)) && (3.12) \\
= \quad & -\frac{1}{N} \sum_{i=1}^{N} (c_{x_i} \log_2 p(1|y_i) + (1 - c_{x_i}) \log_2(1 - p(1|y_i))) && (3.13)
\end{aligned}$$

Die Hilfsfunktion c_x erlaubt ein Aufteilen der Summe in zwei Teilsummen für die beiden möglichen Werte von x (0 und 1). Damit ergibt sich, bis auf das Vorzeichen, genau der Exponent aus Formel 3.6, der nun als Definition für das *kontinuierliche Entropie-Qualitätsmaß* (CE-QM) dient.

$$\begin{aligned}
\text{CE-QM} &= -\log_2(\text{GM-QM}) \\
&= -\frac{1}{N}\Big(\sum_{w\in\mathcal{K}_H} \log_2 p(k \mid w) + \sum_{w\in\mathcal{F}_H} \log_2(1 - p(k \mid w))\Big) \qquad (3.14)
\end{aligned}$$

Diese Formel ist auch unter dem Namen "Cross-Entropy" bekannt. Ein Nachteil dieses Qualitätsmaßes ist die Abhängigkeit von der Qualität des Spracherkenners. Es stellt sich nun die Frage, wie die Ausgabe eines a priori Klassifikators zu wählen ist, daß die Entropie minimiert oder das GM-QM maximiert wird. Die Herleitung wird anhand Formel 3.14 geführt. Mit der Bedeutung für p und t wie bei der Untersuchung des B-QM ergibt sich:

$$\text{CE-QM}(t) = -(p \log_2(t) + (1 - p)(\log_2(1 - t))) \qquad (3.15)$$

Gesucht wird ein Minimum, es muß also $d\text{CE-QM}(t)/dt = 0$ gelten und ein Vorzeichenwechsel von "$-$" nach "$+$" stattfinden.

$$\begin{aligned} \frac{d\text{CE-QM}(t)}{dt} &= -\left(\frac{p}{t} - \frac{1-p}{1-t}\right) \\ &= -\frac{p-t}{t(1-t)} \end{aligned} \tag{3.16}$$

Die Ableitung nimmt für $p = t$ den Wert 0 an, das gilt, wie man leicht überprüfen kann, auch für die (uninteressanten) Sonderfälle $p = 100$ % oder $p = 0$ %.

Es kann $p \neq 100$ % und $p \neq 0$ % angenommen werden. Sei $\epsilon > 0$, aber kleiner als $\frac{1}{2}\min(1-p, p)$. Setzen wir $t = p \mp \epsilon$, so folgt

$$\begin{aligned} &\text{von "links", also } t = p - \epsilon \\ -&\frac{p-(p-\epsilon)}{\underbrace{(p-\epsilon)}_{>0}\underbrace{(1-(p-\epsilon))}_{>0}} > 0 \\ &\text{von "rechts", also } t = p + \epsilon \\ -&\frac{p-(p+\epsilon)}{\underbrace{(p+\epsilon)}_{>0}\underbrace{(1-(p+\epsilon))}_{>0}} < 0 \end{aligned}$$

Daraus ergibt sich, daß das CE-QM für den a priori Klassifikator minimal ist, wenn als Ausgabe $t = p$ gewählt wird. Wegen des engen Zusammenhanges zum geometrischen Mittel muß für das GM-QM der gleiche Wert gewählt werden. Setzt man diesen Wert in Formel 3.15 ein, so wird vom informationstheoretischen Blickpunkt die Entropie H(X) der Quelle berechnet (Abbildung 3.3).

Das NIST-Qualitätsmaß

Vom National Institute of Standards and Technology (NIST) wurde eine "normalisierte Cross-Entropy" als Qualitätsmaß für Vertrauensmesser vorgeschlagen, wir nennen es NIST-QM.

$$\text{NIST-QM} = \frac{H(X) + \frac{1}{N}\left(\sum_{w \in \mathcal{K}_H} \log_2 p(k|w) + \sum_{w \in \mathcal{F}_H} \log_2(1 - p(k|w))\right)}{H(X)} \tag{3.17}$$

dabei ist

$$\begin{aligned} p &= \frac{|\mathcal{K}_H|}{|\mathcal{W}_H|} \\ H(X) &= -(p \log_2 p + (1-p)\log_2(1-p)) \end{aligned}$$

Zusammen mit den oben gefundenen Beziehungen ergibt sich

$$\begin{aligned} \text{NIST-QM} &= \frac{H(X) - \text{CE-QM}}{H(X)} \\ &\approx \frac{H(X) - H(X|Y)}{H(X)} \\ &= \frac{I(X;Y)}{H(X)} \end{aligned}$$

Dabei ist $I(X;Y)$ die Transinformation[5] oder wechselseitige Information (mutual information). Ist der Wert von $I(X;Y) = 0$ so bedeutet dies, daß durch einen Kanal keine Informationen übertragen werden können, weil zwischen einem "gesendeten" und dem korrespondierenden "empfangenen" Zeichen kein Zusammenhang besteht; sie sind unkorreliert. Der maximal erreichbare Wert den $I(X;Y)$ besitzen kann ist $H(X) = I(X;X)$, das bedeutet, daß der Kanal ausreicht, um die gesamte Information der Quelle unverfälscht zu übertragen. Als Folge sollte sich das Qualitätsmaß NIST-QM in einem Wertebereich von 0 bis 1 aufhalten. Dabei ist 1 der maximal erreichbare Wert. Falls ein negativer Wert auftritt ist der Vertrauensmesser schlechter als ein a priori Klassifikator.[6]

Es sei noch auf den bedeutsamen Zusammenhang der Transinformation zur Kullback Leibler Distanz ($D(q\|p) = \sum_{x\in\mathcal{X}} q(x)\log(q(x)/p(x))$) hingewiesen. Die Distanz ist ein Maß für die Gleichheit von Wahrscheinlichkeitsfunktionen. Sind beide Verteilungen identisch, ergibt sich 0. Es gilt die Beziehung $I(X;Y) = D(p(x,y)\|p(x)p(y))$. Gilt $p(x,y) = p(x)p(y)$, so ist dies ebenfalls die Definition für stochastische Unabhängigkeit. Mehr über die Beziehungen bezüglich Entropie, Transinformation und Kullback Leibler Distanz kann in [4] nachgelesen werden.

Das binäre Entropie-Qualitätsmaß

Cox und Rose haben in [5] ein Qualitätsmaß für binäre Vertrauensmesser vorgeschlagen, das ebenfalls auf der Informationstheorie beruht. Der Vertrauensmesser

[5]Übertragungsrate nach Shannon [30], d.h. wieviel der Kanal nach Abzug der Äquivokation überträgt.

[6]Der Vertrauensmesser muß dann eigentlich der Störquelle zugeordnet werden.

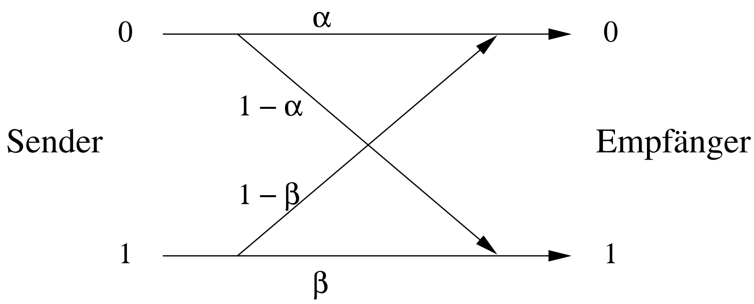


Abbildung 3.4: Binärer Kommunikationskanal

ist wieder Teil des Übertragungskanals, von dem angenommen wird, daß es sich dabei um einen binären Kanal handelt, (Abbildung 3.4). Dabei bezeichnet α die Wahrscheinlichkeit, daß eine 0 auf eine 0 abgebildet wird, β analog für eine 1 als Eingabe in den Kanal. Beide Parameter können einfach aus einer Konfusionsmatrix geschätzt werden. Das von Cox und Rose [5] beschriebene Qualitätsmaß bezeichnen wir als *binäres Entropie Qualitätsmaß* (BE-QM).

$$\text{BE-QM} = \frac{H(X) - H(X|Y)}{H(X)} \times 100\% \quad (3.18)$$

$$= \frac{I(X;Y)}{H(X)} \times 100\% \quad (3.19)$$

dabei ist

$$H(X|Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{\sum_z p(z,x)} \quad (3.20)$$

$$(3.21)$$

Dabei ist $H(X)$ wie in Formel 3.17 definiert und p(x,y) die Wahrscheinlichkeitsverteilung, die durch die Konfusionsmatrix bestimmt wird. Beispielsweise wäre nach Abbildung 3.4 $p(0,1) = (1-\alpha)$.

Präzision, Ausschöpfung und Power

Qualitätsmaße, bei denen ein Vergleich unterschiedlicher Vertrauensmesser nicht einfach ist, die aber eine praktische Bedeutung haben, sind *Präzision* (PRC = precision) und *Ausschöpfung* (REC = recall). Dabei wird betrachtet wie gut die Marken innerhalb einer Klasse vergeben werden, so daß vier Werte (oder Graphen, falls der Vertrauensmesser eine kontinuierliche Ausgabe[7] besitzt), eine Bewertung der Qualität ermöglichen. Die Kennzahlen ergeben sich wie folgt

[7] nicht notwendigerweise eine a posteriori Wahrscheinlichkeit!

$$PRC_k = \frac{|\mathcal{K}_H \cap \mathcal{T}_k|}{|\mathcal{W}_H \cap \mathcal{T}_k|} \tag{3.22}$$

$$REC_k = \frac{|\mathcal{K}_H \cap \mathcal{T}_k|}{|\mathcal{K}_H|} \tag{3.23}$$

$$PRC_f = \frac{|\mathcal{F}_H \cap \mathcal{T}_f|}{|\mathcal{W}_H \cap \mathcal{T}_f|} \tag{3.24}$$

$$REC_f = \frac{|\mathcal{F}_H \cap \mathcal{T}_f|}{|\mathcal{F}_H|} \tag{3.25}$$

Für kontinuierliche Vertrauensmesser können für jede Klasse, korrekt (k), falsch (f), die Werte bestimmt und über der Schwelle oder auch gegeneinander aufgetragen werden. Für verschiedene Anwendungen, beispielsweise Sprecheradaption, kann so ein Arbeitspunkt für die Schwelle bestimmt werden, indem Präzision und Ausschöpfung gegeneinander abgewogen werden. Abbildung 3.5 beispielsweise zeigt einen Graphen für Präzision und Ausschöpfung korrekt erkannter Worte.

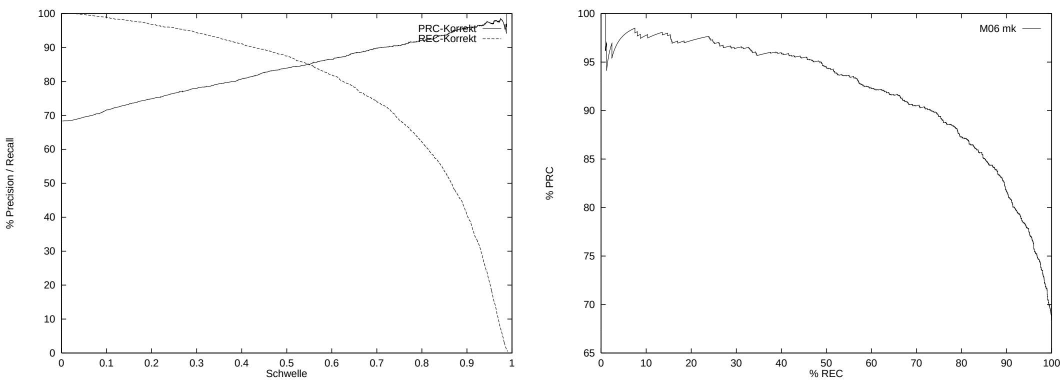


Abbildung 3.5: Präzision und Ausschöpfung für die Klasse korrekt über der Schwelle (links) und gegeneinander (rechts)

Young und Ward [31][32] definieren ein weiteres Qualitätsmaß, genannt Power-QM. Dabei handelt es sich um die Präzision PRC_f bei einer Ausschöpfung $REC_k = 95$ %, das heißt die Schwelle ist so gewählt, daß höchstens 5 % der korrekten Worte als falsch klassifiziert werden. Je mehr korrekte Worte vorhanden sind, und damit im Verhältnis weniger falsche Worte, desto mehr korrekte Worte dürfen als falsch markiert werden. Diese Maß ist speziell auf die Bewertung ausgerichtet wie gut fehlerhafte Worte entdeckt werden, ohne dabei zu viel korrekte Worte abzulehnen.

Jedes der hier vorgestellten Qualitätsmaße besitzt sowohl Vor- als auch Nachteile. Im Rahmen dieser Arbeit konzentrieren wir uns auf das NIST-QM, da hiermit bewertet wird wie gut die Schätzung einer a posteriori Wahrscheinlichkeit möglich ist. Außerdem wird das binäre Qualitätsmaß (B-QM) als Maß für die Klassifikationsleistung angegeben.

Kapitel 4

Der Spracherkenner und die Datenbasis

In diesem Kapitel werden kurz das eingesetzte Janus-2-System und das Janus-3-Programm vorgestellt. Dabei handelt es sich um den Forschungsprototyp eines Spracherkenners, der in Kooperation zusammen vom Institut für Logik, Komplexität und Deduktionssysteme der Universität Karlsruhe und dem Interactive System Lab der Carnegie Mellon University, Pittsburgh entwickelt wird. Anschließend werden die in dieser Arbeit verwendeten Daten beschrieben und nach ihrem Ursprung und ihrer Zusammensetzung aufgeschlüsselt.

4.1 Janus-2-System und Janus-3-Programm

Das Janus-3-Programm stellt Algorithmen und Techniken für die Spracherkennung bereit. Es ist ein reiner Forschungsprototyp, der in die Script-Sprache TCL/TK von Ousterhout [20] eingebettet ist. Dadurch ergibt sich die Möglichkeit, die implementierten Techniken sehr einfach konfigurieren und kombinieren zu können. Weiter ergibt sich eine hohe Portabilität und die einfache Möglichkeit graphische Benutzerführungen zu erhalten. Aufgrund der durchdachten Einbettung ist es möglich, mit verhältnismäßig geringem Aufwand eigene Erweiterungen zu integrieren. Zu einem gut funktionierenden Spracherkenner gehört aber nicht nur ein gutes Programm, besonders wichtig sind die verwendeten Modelle, die im Training gefunden werden. Das im Rahmen dieser Arbeit eingesetzte Janus-System (Janus-2) wurde für die Verbmobil-Evaluation des BMFT im Juni 1995 verwendet und erzielte dort mit einer Wortakkuratheit von 71,4 % von allen Teilnehmern das beste Ergebnis.

Das Janus-2-System verwendet Gausssche Mischverteilungen (mixture-gaussian) für die HMM mit der skalierbaren Möglichkeit Parameter zu teilen (semikontinuierliche HMM). Für die HMM-Zustände werden 1677 kontextabhängige Subtriphone mit gemeinsamen 1338 Mischverteilungen eingesetzt. Als Signalvorverarbeitung werden Melscale-Filterbank-Koeffizienten [33] mit einer Framerate von 10 ms und deren delta-Koeffizienten verwendet. Zusätzlich wird die Signalenergie und Nulldurchgangsrate sowie der peak-to-peak-Wert benutzt. Insgesamt ergibt sich ein 37-dimensionaler Eingabevektor, der durch eine lineare Diskriminanzanalyse [9] auf einen 32-dimensionalen Vektor reduziert wird. Dieser wird wiederum in zwei 16-dimensionale Datenströme aufgespalten, von denen der für die Erkennung bedeutendere verwendet wird. Normalerweise werden beide Datenströme für die Erkennung herangezogen, was den Rechenaufwand verdoppelt und die Erkennungsleistung nur unwesentlich verbessert.

Um zu einer Hypothese zu gelangen, werden mehrere Suchschritte durchgeführt in denen der Suchraum strukturiert wird. Um rasch eine Menge an aussichtsreichen Hypothesen zu bekommen, wird znächst eine *Baumsuche* durchgeführt, bei der die HMM-Modelle der Worte baumartig strukturiert sind. Hierbei läßt sich ein Sprachmodell nicht richtig integrieren. Anhand des Ergebnisses wird eine *flache Suche* durchgeführt (jedes Wort besitzt dabei ein eigenes HMM-Modell), bei der die genauen Zeitgrenzen der Worte bestimmt werden und ein Bigramm-Sprachmodell eingesetzt wird. Aus dem Ergebnis dieser Suche wird ein Worthypothesengraph berechnet, der die Verwendung auch komplexer Sprachmodelle erlaubt. Eingesetzt wird ein Bigramm-Sprachmodell. Die Größe des Vokabulars beträgt 3439 Worte (inklusive Müllworte). Eine ausführliche Beschreibung des Janus-2-Systems ist in [37] zu finden.

4.2 Verbmobil-Datenbasis

Eine wichtige Voraussetzung für aussagekräftige Ergebnisse ist, daß genügend Datenmaterial für die Untersuchung und Bewertung zur Verfügung steht. Trainingsdaten dienen zur Untersuchung von Merkmalen und zur Konstruktion oder zum Training verschiedener Klassifikatoren; die Kreuzvalidierungsstichprobe dient zur Auswahl des Klassifikators, der für eine Bewertung auf der Evaluationsmenge benutzt wird. Da für das akustische Training eines Spracherkenners und das Training des Sprachmodells selbst große Datenmengen benötigt werden, sind frei verfügbare Daten normalerweise nicht in ausreichender Mengen vorhanden. Die Daten für die Untersuchungen dürfen aus methodischen Gründen nicht für das Training des Spracherkenners verwendet worden sein. Trotzdem stand für die Untersuchung eine ausreichende Menge an Daten zur Verfügung, die im Rahmen des Verbmobil-Projekts[1] an vier verschiedenen Orten Deutschlands (Kiel, Bonn,

[1] German Spontaneous Scheduling Task

Karlsruhe, München) gesammelt wurden. Eine Beschreibung der Datensammlung findet sich in Abschnitt 2.2.

Weil die Evaluationsmenge Teilmenge der Verbmobil-Evaluation ist, an der das Janus-2-System teilgenommen hat, enthält sie keine Buchstabierungen. Um eine ausreichend große Datenmenge für die Trainingsmenge und die Kreuzvalidierungsstichprobe benutzen zu können mußte aber in Kauf genommen werden, daß die Äußerungen Buchstabierungen enthalten. Da der gleiche Spracherkenner wie für die Evaluation verwendet wird sind Buchstabierungen nicht vorgesehen. Die deutliche Diskrepanz zwischen den Datenmengen stellt insofern eine Schwierigkeit dar, da ein Vertrauensmesser möglicherweise die a posteriori Wahrscheinlichkeiten dann nicht angemessen schätzen kann.

Bei der Aufteilung der Datenmengen in Trainingsmenge und Kreuzvalidierungsstichprobe wurde darauf geachtet, daß es keinen Überlapp der Spender gibt, um die Sprecherunabhängigkeit des Vertrauensmessers nachzuweisen. Das ergab Schwierigkeiten, da die Anzahl und Länge der Äußerungen der einzelnen Spender stark variiert und eine möglichst gleichmäßige Aufteilung bezüglich der Aufnahmeorte gewünscht wurde. Als Folge ergab sich eine ungleichmäßige Aufteilung bezüglich den Aufnahmen weiblicher und männlicher Sprecher zwischen der Trainingsmenge und der Kreuzvalidierungsstichprobe. Tabelle 4.1 zeigt die Zusammensetzung der verwendeten Daten aufgeschlüsselt nach Aufnahmeorten.

Menge	Ort	Sprecher	Aufnahmen	Dauer (min.)
Training	Kiel	4	178	21
	Bonn	15	140	18
	Karlsruhe	8	190	23
	München	19	277	39
Summe		46	785	101
Kreuzvalidierung	Kiel	2	107	13
	Bonn	9	68	10
	Karlsruhe	3	130	17
	München	12	161	21
Summe		26	466	61
Evaluation	Kiel	6	73	6
	Bonn	24	60	6
	Karlsruhe	9	72	6
	München	29	60	6
Summe		68	265	24

Tabelle 4.1: Datenmengen: Aufteilung nach Sammelorten

Menge	Dauer min.	Aufnahmen weiblich	männlich	Worte insges.	OOV-Worte	Buchst.
Training	101	271	514	14914	1263	509
Kreuzvalidierung	61	282	184	9029	622	288
Evaluation	24	100	165	3758	191	0

Tabelle 4.2: Datenmengen: Aufnahmedauer, Sprechergeschlecht, Worte insgesamt

Tabelle 4.2 faßt die Daten aus Tabelle 4.1 zusammen und gibt die Anzahl der Worte (ohne Müllworte) und die Zahl der OOV-Worte an. Buchstabierungen werden zwar ebenfalls zu den OOV-Worten gezählt, sind aber nochmals explizit angegeben. Es ergeben sich für die drei Mengen ungefähr gleichgroße OOV-Raten von ungefähr 3 bis 5 %, wenn man die Buchstabierungen von den OOV-Worten abzieht.

Mit dem Spracherkenner Janus wurden für diese Mengen Hypothesen berechnet, deren Worte bewertet werden sollen. Tabelle 4.3 gibt an, wieviel Worte die Hypothesen der einzelnen Mengen enthalten.

Durch den Wortübergangsstrafterm (WP) ist es möglich, den Spracherkenner so einzustellen, daß er eine minimale Anzahl an Einfügungen erzeugt. Dadurch erhöht sich im allgemeinen der Korrektanteil innerhalb der Hypothese; aber die Wortakkuratheit wird schlechter, da die Zahl der Löschfehler zunimmt. Für einen so eingestellten Spracherkenner ist es zwar möglicherweise einfacher einen Vertrauensmesser mit guten Leistungen zu erhalten, aber der Spracherkenner selbst würde als schlecht angesehen werden! Primär muß aber der Spracherkenner gute Ergebnisse liefern.

Menge	Worte	korrekt	falsch	% korrekt
Training	14906	10348	4558	69,4 %
Kreuzvalidierung	9002	6645	2357	73,8 %
Evaluation	3839	2623	1216	68,3 %

Tabelle 4.3: Datenmengen: Korrektraten in den Hypothesen

Bis auf die Evaluationsmenge entsprechen sich die Zahlen der Worte in den Hypothesen und den Transkriptionen fast. Das bedeutet, daß die Zahl der Einfüge- und Löschfehler fast gleich groß ist. Der Wortübergangsstrafterm des Spracherkenners ist also für die Trainingsmenge und die Kreuzvalidierungsstichprobe fast optimal eingestellt.

In Tabelle 4.4 sind die Fehler weiter aufgeschlüsselt. Buchstabierungsfehler sind dabei Teilmenge der OOV-Fehler und diese sind eine Teilmenge der Ersetzungsfehler.[2] Weil Löschfehler nur in Referenzsätzen auftreten, wurden sie in der Tabelle abgesetzt eingetragen. Die letzte Spalte gibt die Wortakkuratheit (WA) der einzelnen Teilmengen an. Wie aus dem Anteil korrekt (Tabelle 4.3) und der Wortakkuratheit (WA Tabelle 4.4) zu ersehen ist, scheinen die Teilmengen für den Spracherkenner unterschiedlich schwer zu erkennen zu sein. Wird die Trainingsmenge und die Kreuzvalidierungsstichprobe zusammengenommen, ergibt sich eine Wortakkuratheit von 65,7 %.

Menge	Lösch-fehler	Einfüge-fehler	Ersetzungs-fehler	OOV-Fehler	Buchst.-fehler	WA
Training	839	831	3727	1144	477	63,8
Kreuzvalidierung	461	434	1923	573	275	68,8
Evaluation	205	280	930	137	0	62,1

Tabelle 4.4: Aufteilung der Fehler in den Datenmengen

[2]Ein OOV-Wort kann auch bei einer Löschung auftreten, dies wird aber hier nicht als ein OOV-Fehler angeshen. Welche Schwierigkeit sich weiter mit OOV-Worten ergeben ist in Abschnitt 3.1 erläutert.

Kapitel 5

Untersuchte Wissensquellen und abgeleitete Merkmale

In diesem Kapitel werden Merkmale beschrieben, die daraufhin untersucht wurden, wie gut sie sich für die Bewertung der Glaubwürdigkeit $p(Korrekt|w)$ eines Wortes w eignen. Zunächst werden wortassoziierte Merkmale betrachtet. Dazu zählen wir Merkmale, die nur die Worte der Hypothese benötigen. Anschließend wird die Zeitdauer der Hypothesenworte untersucht, da diese bei der Berechnung der Hypothese unberücksichtigt bleibt. Danach wird betrachtet, welche Möglichkeiten sich zur Bewertung der Glaubwürdigkeit aus der akustischen Ähnlichkeit ergeben. Abschließend werden Möglichkeiten betrachtet, Unsicherheit im Suchraum zu messen. Um die Qualität eines Merkmals einzuschätzen werden Korrelationskoeffizienten [9] berechnet. Da die Merkmale durchaus mit den verschiedenen Fehlerklassen unterschiedlich stark korreliert sein können, werden zum Koeffizienten für Gesamtfehler noch die Werte für Einfügefehler und OOV-Fehler angegeben. Alle Werte wurden, wenn nicht anders angegeben, auf der vereinigten Trainingsmenge und Kreuzvalidierungsstichprobe berechnet.

5.1 Wortassoziierte Merkmale

Für die Bestimmung der Glaubwürdigkeit einer Hypothese wurden zunächst wortassoziierte Merkmale untersucht. Bei diesen Merkmalen werden lediglich die Worte der Hypothesen ohne ihre konkreten akustischen Eigenschaften, beispielsweise Zeitdauer, betrachtet. Die untersuchten wortassoziierten Wissensquellen sind:

1. Wortidentität
2. Wortlänge
3. Sprachmodell
4. Häufigkeit des Wortes im akustischen Training.

5.1.1 Wortidentität

Zunächst soll das Merkmal Wortidentität näher betrachtet werden. Unter Wortidentität verstehen wir dabei das Lexem, so wie es im Vokabular des Spracherkenners auftritt. Um eine sichere Schätzung der Wahrscheinlichkeit $p(Korrekt|w)$ für ein Wort w durchführen zu können, wird für jedes Wort w, das in einer Hypothese vorkommen kann, eine hinreichend große Menge an Beispielen benötigt. Für einen Spracherkenner mit einem großen Wortschatz ist solch eine umfangreiche Datenmenge aber aufwendig und teuer zu erhalten.

Die 20 häufigsten Worte aus den Hypothesen der Trainingsmenge sind in Tabelle 5.1 mit der Anzahl ihres Auftretens, der Fehlerrate, der Rate an Einfügefehler, und der OOV-Fehlerrate angegeben.

Es gibt große Unterschiede in der Fehlerrate, beispielsweise zwischen dem häufigsten und zweithäufigsten Wort. Weiter fällt auf, daß «'s» das einzige Wort in der Tabelle ist, bei dem eine Fehlerrate größer 50 % (61,4 %) auftritt. Fast die Hälfte der Fehler gehen in diesem Fall darauf zurück, daß dieses Wort in den Hypothesen als Einfügung erscheint.[1]

Zur Beurteilung der Qualität des Merkmals *Wortidentität* wurde folgendes Experiment durchgeführt: Alle Worte, die mindestens 40mal in der Trainingsmenge vorgekommen sind (85 Worte), oder die mindestens 10mal falsch erkannt wurden (95 Worte), wurden in eine Tabelle mit ihrer Korrektrate (= 100 % - Fehlerrate) aufgenommen. Diese Tabelle enthält insgesamt 112 Worte und deckt 63,5 % aller Worte der Trainingsmenge ab. Alle Worte, die nicht in dieser Tabelle enthalten sind, wurden in einer Restklasse zusammengefaßt und ihre Korrektrate (70,2 %) bestimmt. Die Tabelle wurde benutzt, um jedem Wort aus den Hypothesen der Kreuzvalidierungsstichprobe ein Attribut für die Glaubwürdigkeit zuzuordnen. Beispielsweise wird dem Wort «ich» eine Glaubwürdigkeit von 82,8 % zugewiesen, dem Wort «das» 55,1 % und dem Wort «die» 51,5 %. Ein Wort, das der Restklasse zugeordnet ist, beispielsweise «Hans», erhält dann eine Glaubwürdigkeit von 70,2 %.

[1] Wie sich in Abschnitt 5.1.2 zeigen wird, hängt dies mit der Länge des Wortes zusammen.

Lexem	Anzahl	Gesamt-FR	Einfüge-FR	OOV-FR
ich	552	18,8 %	2,9 %	6,0 %
das	397	45,8 %	5,0 %	3,0 %
ja	380	22,4 %	7,9 %	5,3 %
dann	306	46,1 %	10,8 %	2,9 %
wir	267	28,5 %	6,7 %	3,7 %
am	267	27,7 %	13,9 %	3,7 %
und	200	22,5 %	7,5 %	2,5 %
da	192	42,2 %	18,2 %	3,1 %
mir	189	27,5 %	5,3 %	2,1 %
ist	181	24,9 %	3,3 %	5,5 %
bis	177	13,0 %	1,7 %	2,3 %
der	174	32,2 %	6,9 %	5,2 %
Uhr	173	30,1 %	11,0 %	5,8 %
den	168	26,8 %	7,7 %	2,4 %
’s	153	61,4 %	27,5 %	3,9 %
Sie	151	28,5 %	2,6 %	7,9 %
gut	148	16,2 %	2,7 %	6,1 %
noch	141	31,2 %	6,4 %	2,8 %
die	134	48,5 %	9,7 %	22,4 %
also	125	20,8 %	1,6 %	3,2 %

Tabelle 5.1: Fehlerrate über die Wortidentität

Tabelle 5.2 zeigt die Korrelation zur Fehlerrate für die Trainingsmenge und die Kreuzvalidierungsstichprobe. Ein Klassifikator, der mit dieser Tabelle arbeitet, erreicht eine Klassifikationsleistung auf der Kreuzvalidierungsstichprobe, wie sie in Tabelle 5.3 angegeben ist. Aufgrund des geringen Datenmaterials gibt es auch einige Worte, die zu 100 % korrekt oder falsch erkannt wurden. Für diese Worte in der Tabelle wurde der Eintrag um 1 % modifiziert, um den Fall einer totalen Fehlklassifikation zu vermeiden, der sich beim NIST-QM besonders negativ auswirkt.

	Kreuzvalidierungsstichprobe	Trainingsmenge
Gesamtfehler	-25,9 %	-28,0 %

Tabelle 5.2: Korrelation: Wortidentität

Obwohl eine relativ große Korrelation zum Fehler besteht, erscheint das Merkmal Wortidentität für sich genommen nicht besonders stark zu sein, betrachtet

Qualitätsmaß	Bewertung	Baseline
NIST-QM	0.051	0.000
B-QM	74,7 %	73,8 %

Tabelle 5.3: Bewertung des Klassifikators: Wortidentität

man die Werte in der Tabelle 5.3. Allerdings ergibt sich bereits eine Verbesserung von 0,9 % absolut (3,4 % relativ) in der Bewertung durch B-QM gegenüber der Baseline, was für ein Einzelmerkmal mit diesem einfachen Ansatz beachtlich ist und andeutet, daß wesentlich mehr möglich ist. Beispielsweise haben Young und Ward [32], [31] durch die Verwendung einer speziellen Einteilung ihrer Datenmenge, basierend auf der Wortidentität, eine deutliche Leistungsverbesserung ihres untersuchten Merkmals, einer normalisierten Wort-Score, erreicht. Ein ähnlicher Effekt konnte von Qiu [21] auch festgestellt werden. Die Bildung sogenannter Anti-Wortmodelle, wie sie beispielsweise von Rose, Lleida, Sukkar und Rahim [29], [13], [34], [35], [26] für ein diskriminatives Training verwendet werden, beruhen ebenfalls auf der Wortidentität.

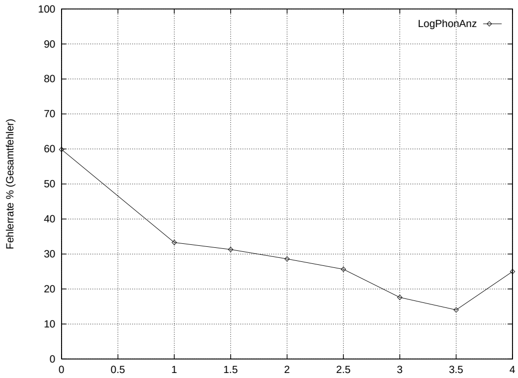
5.1.2 Wortlänge

Ein Merkmal, das aus der Wortidentität abgeleitet werden kann, ist die Wortlänge. Cox und Rose [5] haben dieses Merkmal ebenfalls untersucht und die Länge eines Wortes als die Anzahl seiner Phoneme definiert (PhonAnz). Tabelle 5.4 zeigt die Korrelation von Phonemanzahl und Fehlerrate. Eine deutliche Beziehung zwischen der Länge und der Fehlerrate ist zu sehen. Je länger ein Wort ist, desto sicherer wurde es richtig erkannt. Dies beruht im wesentlichen darauf, daß längere Worte längere HMM ergeben, und somit mehr Sprachsignal mit dem Modell übereinstimmen muß.

	Gesamt-FR	Einfüge-FR	OOV-FR
PhonAnz	-10,3 %	-9,8 %	-0,5 %
LogPhonAnz	-11,8 %	-12,1 %	-1,0 %

Tabelle 5.4: Korrelation: Wortlänge

Wie aus der Korrelationstabelle auch entnommen werden kann, wird die Korrelation verbessert, wenn die Länge logarithmiert (LogPhonAnz) wird. Abbildung 5.1 zeigt den Verlauf der Fehlerrate über die logarithmierte Länge, dabei wurde als Basis des Logarithmus 2 gewählt. Die Länge des Wortes korreliert stark mit den Einfügefehlern, es ist jedoch so gut wie keine Korrelation zu OOV-Fehlern vorhanden. Je kürzer ein HMM-Modell eines Wortes ist, desto wahrscheinlicher ist es, daß eine zufällig gut passende Beobachtungssequenz in einer Äußerung auftreten kann. Dies führt dann zu Einfügefehlern. Die Länge eines HMM-Modells ist von der Anzahl und Art der Phoneme des dazugehörenden Wortes bestimmt (vgl. Abschnitt 2.5). Ein OOV-Wort ist in der Regel ein längeres Wort, das in kürzere zerfällt. Als Beispiel sei wieder «füreinander» angegeben, das in drei kurze Worte zerfallen ist, wovon zwei als Einfügefehler gezählt wurden (vgl. Abschnitt 3.1). Die Worte, in die ein OOV-Wort bei der Erkennung zerfallen kann, können sehr unterschiedlich lang sein.



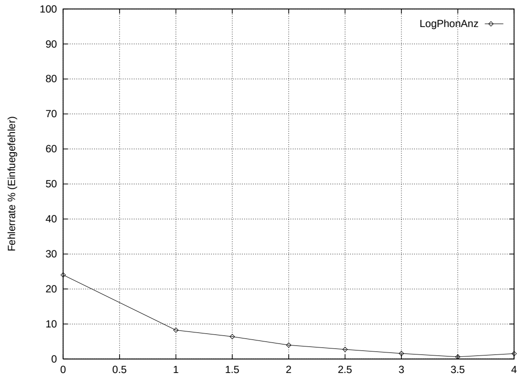


Abbildung 5.1: Gesamtfehler (links) und Einfügefehler (rechts) über die logarithmierte Wortlänge

5.1.3 Sprachmodell

In [7] berichten Eide, Gish, Jeanrenaud und Mielke, daß sie einen Zusammenhang zwischen der Wahrscheinlichkeit, daß ein verwendetes Sprachmodell die betrachtete Hypothese liefert, und Erkennungsfehlern festgestellt haben. Ein anderes von ihnen [7] untersuchtes Merkmal ist, wie häufig ein Trigramm im Training aufgetreten ist. Dabei wurde ebenfalls ein Zusammenhang zum Auftreten von Fehlern festgestellt. In diese Arbeit wurde eine Variante betrachtet, die feststellt, ob eine Interpolation bei der Berechnung der Sprachmodellwahrscheinlichkeit durchgeführt werden mußte.

Dabei wurden zu einem betrachteten Wort die Vorgängerworte in der Hypothese bestimmt und untersucht, ob eine Schätzung für diese Wortfolge existiert. Wenn dies nicht der Fall war, wurde die Wortfolge um ein Wort verkürzt und wieder im Sprachmodell nachgesehen. Dies wurde solange wiederholt, bis eine Schätzung für die verkürzte Wortfolge im Sprachmodell vorhanden war. Spätestens bei einer Länge von einem Wort ist immer eine Schätzung möglich, da jedes

Wort, das hypothetisiert werden kann, als Monogramm im Sprachmodell enthalten ist. Die um 1 verringerte Länge der verbliebenen Wortfolge wird als Merkmal genommen (SM-NGRAM). Da für diese Arbeit ein Sprachmodell mit Bigrammen vorlag, ergeben sich dadurch nur die Werte 0 und 1. Tabelle 5.5 zeigt die Korrelation zum Fehler. Dabei zeigt sich, daß insbesondere eine starke Korrelation zu OOV-Fehlern vorliegt. Soll ein OOV-Wort wie beispielsweise «füreinander» erkannt werden, passen häufig die akustischen Modelle sehr viel besser als die Sprachmodelle. Unwahrscheinliche Wortfolgen kommen selten oder gar nicht in den Trainingsmengen für das Sprachmodell vor, weshalb eine Interpolation durchgeführt werden muß.

	Gesamt-FR	Einfüge-FR	OOV-FR
SM-NGRAM	-19,6 %	-0,7 %	-14,3 %

Tabelle 5.5: Korrelation: Sprachmodell

In Tabelle 5.6 ist die Verteilung der Fehlerrate für die gesamte Trainingsmenge angegeben. Ein Klassifikator, der allein auf diesem Merkmal beruht, kann bei dieser Verteilung der Fehlerrate bei einer binären Klassifikation keine Ablehnung generieren, ohne sich dabei nach dem hier geforderten Qualitätsmaß für binäre Klassifikation zu verschlechtern.

SM-NGRAM	1	0
Gesamt-FR	23,0 %	42,6 %
Einfüge-FR	4,2 %	7,9 %
OOV-FR	4,8 %	12,7 %

Tabelle 5.6: Fehlerrate: Sprachmodell

5.1.4 Häufigkeit im akustischen Training

In [7] wurde ebenfalls berichtet, daß zwischen der Häufigkeit eines Wortes im akustischen Training oder Sprachmodell-Training und der Fehlerrate des Wortes ein Zusammenhang besteht. In dieser Arbeit wurde dieses Phänomen für den Fall des akustische Trainings untersucht (AnzATrain, LogAnzATrain). Hierzu wurden die Worte in den Transkriptionen der akustischen Trainingsmenge gezählt und eine Tabelle aufgestellt. Jedes Wort aus der gesamten Trainingsmenge erhielt die Häufigkeit seines Auftretens in der akustischen Trainingsmenge als Attribut. Die Korrelation dieses Merkmals mit der Fehlerrate wurde untersucht und ist in

Tabelle 5.7 dargestellt. Eine deutliche Korrelation zu OOV-Fehlern kann dabei festgestellt werden.

Es zeigt sich an dieser Stelle besonders deutlich, daß eine geringe lineare Korrelation nicht bedeutet, daß ein Merkmal zur Fehleraufdeckung ungeeignet ist. Wie in Tabelle 5.7 zu sehen ist, kann durch eine nicht lineare Transformation (in diesem Fall: log()) ein möglicher Zusammenhang zwischen Merkmal und Fehlerrate stärker hervorgehoben werden. Allgemein gilt aber, daß eine starke Korrelation auf ein geeignetes Merkmal schließen läßt.

	Gesamt-FR	Einfüge-FR	OOV-FR
AnzATrain	-0,4 %	-0,3 %	-0,8 %
LogAnzATrain	-10,7 %	-0,3 %	-16,2 %

Tabelle 5.7: Korrelation: Häufigkeit im akustischen Training

Abbildung 5.2 zeigt die Fehlerrate über die logarithmierte Häufigkeit im Training, als Basis des Logarithmus wurde 2 gewählt.

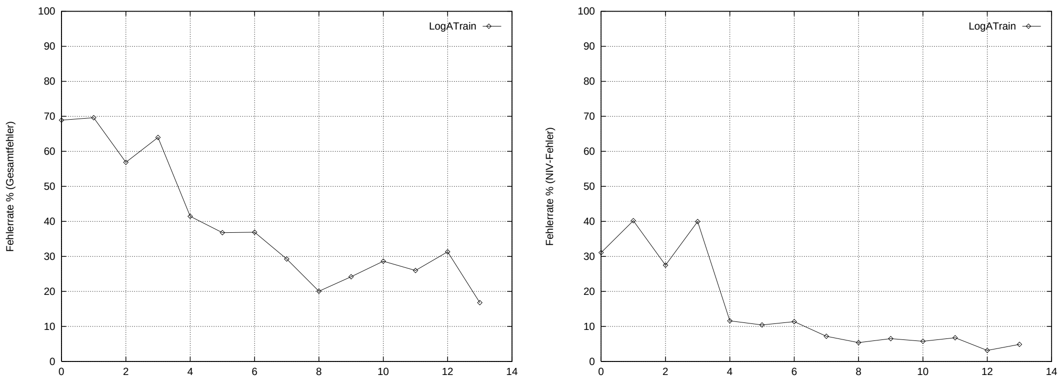


Abbildung 5.2: Gesamtfehler (links) und OOV-Fehler (rechts) über logarithmische Häufigkeit im akustischen Training

Deutlich ist zu erkennen, daß Worte, die selten im Training waren, sehr viel schlechter erkannt werden. Zwei Ursachen sind dafür denkbar. Zum einen, daß für diese Daten eine korrekte Schätzung mit dem Sprachmodell[2] nicht möglich ist, zum anderen, daß die akustischen Modelle für seltene Worte nicht genügend generalisieren. Unabhängig davon welche Ursachen zum Ansteigen der Fehlerrate führen ist dieses Merkmal für eine Bewertung der Glaubwürdigkeit eines Wortes nützlich. Das Ansteigen der Fehlerrate ab dem Wert 8 (256-maliges Vorkommen eines Wortes) beruht vermutlich auf dem verstärkten Auftreten kurzer Worte, die

[2]vergleiche hierzu die Korrelationstabelle 5.9

im allgemeinen eine große Häufigkeit besitzen und schlechter erkannt werden, wie in Abschnitt 5.1.2 festgestellt, beispielsweise «das». Bemerkenswert ist, daß das Wort «ich», verglichen mit dem zweithäufigsten Wort mehr als doppelt so oft in der akustischen Trainingsmenge auftritt. Das führt dazu, daß der am weitesten rechts liegende Datenpunkt allein von diesem Wort gebildet wird.

5.1.5 Bewertung der wortassoziierten Merkmale

In Tabelle 5.8 sind die Merkmale, die am besten mit dem Fehler korrelieren, zusammengestellt. Die Korrelation der Wortidentität wird nicht angegeben, da das Merkmal auf den Trainingsdaten optimal bestimmt wurde und damit ein Vergleich mit den übrigen Korrelationen nicht mehr zulässig ist. Die größte Korrelation sowohl zum Gesamtfehler als auch zum OOV-Fehler ist durch das Sprachmodell (SM-NGRAM) gegeben. Einfügefehler dagegen besitzen eine deutliche Korrelation zur Wortlänge (LogPhonAnz).

	Gesamt-FR	Einfüge-FR	OOV-FR
SM-NGRAM	-19,6 %	-0,7 %	-14,3 %
LogPhonAnz	-11,8 %	-12,1 %	-10,4 %
LogAnzATrain	-10,7 %	-0,3 %	-16,2 %

Tabelle 5.8: Korrelation: Wortassoziierte Merkmale und Fehler

Positiv ist, daß die beiden Merkmale SM-NGRAM und LogPhonAnz zueinander nur wenig korrelieren, wie aus Tabelle 5.9 entnommen werden kann. Dies läßt vermuten, daß eine kombinierte Verwendung in einem Klassifikator eine Verbesserung ergeben könnte.

	SM-NGRAM	LogPhonAnz	LogAnzATrain
SM-NGRAM	100,0 %	-5,3 %	17,2 %
LogPhonAnz	-5,3 %	100,0 %	54,5 %
LogAnzATrain	17,2 %	54,5 %	100,0 %

Tabelle 5.9: Korrelation: Wortassoziierte Merkmale untereinander

5.2 Zur Sprechgeschwindigkeit assoziierte Merkmale

Es ist ein bekanntes Phänomen, daß für unterschiedliche Sprecher deutlich unterschiedliche Erkennungsleistungen auftreten können. Stark sprecherabhängig ist die Sprechgeschwindigkeit, die nicht konstant bleibt, sondern situationsbedingten Variationen unterliegt. Eide et al [7] haben festgestellt, daß insbesondere Äußerungen von 'Schnellsprechern' schlechter erkannt werden. In einem von Kemp durchgeführten Experiment konnte ebenfalls ein Zusammenhang zwischen Sprechgeschwindigkeit und Erkennungsleistung nachgewiesen werden. In diesem Experiment wurde für alle transkribierten Worte der akustischen Trainingsmenge die mittlere Zeitdauer bestimmt. Die Berechnungen wurden auch für jedes Phonem durchgeführt. Die in diesem Experiment gewonnenen Daten wurden für die Berechnung einiger Merkmale in diesem Kapitel zur Verfügung gestellt. Da in einer späteren Anwendung die korrekte Äußerung (Transkription) nicht zur Verfügung steht, kann die Sprechgeschwindigkeit nur mit der Hypothese bestimmt werden; diese wird dann estimierte Sprechgeschwindigkeit genannt. Mit dieser Zeiteinteilung lassen sich nun verschiedene Merkmale untersuchen, die mit der Sprechgeschwindigkeit zusammenhängen:

- Wortstreckung und Wortstauchung
- Schwankung der estimierten Sprechgeschwindigkeit

5.2.1 Wortstreckung und Wortstauchung

Wir definieren Wortstreckung als den Quotienten aus der Zeitdauer eines hypothetisierten Wortes (Lex) und der erwartete mittleren Zeitdauer eines Wortes (Lexem). Beträgt beispielsweise die mittlere Zeitdauer für das Wort «das» 200 ms und die Dauer des Zeitsegments in der Hypothese 150 ms, so ergibt sich als Steckungsfaktor $\frac{150ms}{200ms} = 0,75$. Der Kehrwert der Wortstreckung wird als Wortstauchung definiert. In obigem Beispiel entspricht dies einer Stauchung des Zeitsegments um den Faktor 1,33.[3] Ist das betrachtete Wort der Hypothese identisch mit der tatsächlichen Äußerung und stimmen die gefundenen Wortgrenzen mit den tatsächlichen überein, läßt sich die Stauchung als relative Sprechgeschwindigkeit ansehen.

Für die Bestimmung der erwarteten Wortdauer wurden drei Ansätze verfolgt.

[3]Die Unterscheidung zwischen Wortstauchung und Wortstreckung wird im wesentlichen deshalb getroffen, weil $\frac{1}{x}$ eine nichtlineare Funktion ist. Solche Funktionen können die Korrelation deutlich beeinflussen.

1. Im Deutschen beträgt die mittlere Dauer eines Phonems ungefähr 50 ms. Anhand eines Wörterbuchs mit phonetischer Umschrift wird die Anzahl der Phoneme eines betrachteten Wortes aus der Hypothese bestimmt. Durch Multiplikation erhält man die erwartete Wortlänge. Beispielsweise besteht ≪das≫ aus drei Phonemen, die erwartete Wortlänge beträgt somit 150 ms.

2. Statt anzunehmen, daß alle Phoneme die gleiche Länge besitzen, berechnet man die erwartete Wortlänge aus einer Tabelle, die die mittlere Phonemdauer für jedes Phonem enthält. Anhand des Wörterbuches werden die einzelnen Phoneme bestimmt. Die erwartete Wortlänge ergibt sich aus der Summe der einzelnen Phonemlängen. Für das obige Beispiel ergibt sich mit dieser Tabelle für die drei Phoneme des Wortes ≪das≫ eine erwartete Länge von $(54,5\text{ms}) + (64,6\text{ms}) + (91,5\text{ms}) = 210,6\text{ms}$.

3. Anstatt davon auszugehen, daß die mittlere Länge eines bestimmten Phonems in allen Worten gleichgroß ist, ist es theoretisch auch möglich die erwartete Länge für jedes Wort separat zu bestimmen. In der Praxis ist dies für alle Worte allerdings nicht durchführbar. Für Worte, für die genügend akustisches Material zur Verfügung steht, ist es aber durchaus möglich eine Tabelle der erwarteten Wortlänge zu erstellen. Bei Worten, die nicht darin enthalten sind, muß auf eine der oben beschriebenen Varianten zurückgegriffen werden. Die mittlere Wortlänge für unser Beispielswort ≪das≫ beträgt in der akustischen Trainingsmenge 191,6 ms.

Anhand der in der Hypothese gefundenen Zeitdauer eines Wortes lassen sich nun folgende konkrete Merkmale berechnen. Diese Merkmale werden für die Wortstauchung WStauch1, WStauch2 und WStauch3 genannt, wobei die Ziffer die obige Variante angibt, mit der die erwartete Wortlänge berechnet wird. Analog bezeichnen wir Merkmale, die sich aus der Wortstreckung ergeben mit WStreck1, WStreck2 und WStreck3. Falls bei der Berechnungsvariante 3 ein Wort nicht in der Tabelle enthalten ist, wird Variante 2 zur Berechnung der erwarteten Wortlänge verwendet.

Für die genannten Merkmale wurde auf der gesamten Trainingsmenge die Korrelation zum Fehler bestimmt; sie ist in Tabelle 5.10 angegeben.

Es fällt auf, daß die Merkmale der Wortstreckung deutlich stärker mit dem Fehler korrelieren als die der Wortstauchung. Vor allem gilt das für OOV-Fehler. Der Grund ist, das insbesondere Buchstabierungen gedehnte Phoneme enthalten. WStreck3 besitzt eine Korrelation von 20,9 % mit dem Auftreten eines Buchstabens in der gesamten Trainingsmenge. Erwartungsgemäß zeigt das Merkmal WStreck3 auch die stärkste Korrelation in der Tabelle, was auf die Ausnutzung der Wortidentität bei der Berechnung dieses Merkmals zurückzuführen ist.

	Gesamt-FR	Einfüge-FR	OOV-FR
WStauch1	-2,4 %	-3,6 %	-7,9 %
WStreck1	3,5 %	5,8 %	9,4 %
WStauch2	-2,4 %	-2,7 %	-8,7 %
WStreck2	5,0 %	6,0 %	10,1 %
WStauch3	-1,3 %	-1,6 %	-8,2 %
WStreck3	6,6 %	6,0 %	11,5 %

Tabelle 5.10: Korrelation: Wortstreckung und Wortstauchung

Abbildung 5.3 (links) zeigt die Gesamtfehlerrate über den Merkmalen WStauch1, WStauch2 und WStauch3. Es zeigt sich, daß die lineare Korrelation nicht die wirklichen Zusammenhänge beschreibt, da ein deutlicher nichtlinearer Zusammenhang vorhanden ist. Wie aus der rechten Abbildung abgelesen werden kann, trägt dabei die OOV-Fehlerrate zu einem deutlicher Anstieg der Gesamtfehlerrate bei gedehnten Worten (Stauchung < 1) bei.

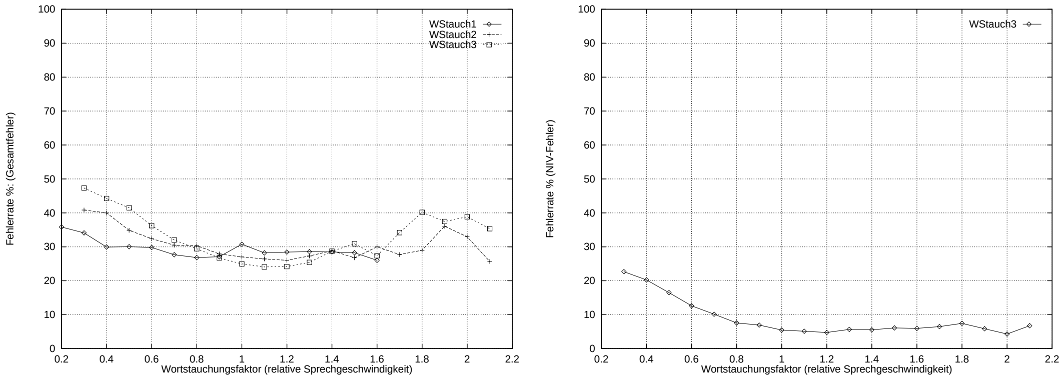


Abbildung 5.3: Gesamtfehler (links) und OOV-Fehler (rechts, nur WStauch3) über die Wortstauchung

Beim Vergleich der drei Kurven in der linken Abbildung zeigt sich wieder, daß die wortbasierte Längenestimierung besser ist als die phonembasierte. Das Minimum der Fehlerrate für WStauch3 liegt tiefer als für die beiden anderen Merkmale, und steigt für größere Abweichungen von der erwarteten Wortlänge stärker an.

Eide, Gish, Jeanrenaud und Mielke [7] verwenden als Maß für die Sprechgeschwindigkeit die Anzahl Phoneme je Sekunde. Diese Definition ist zum Merkmal WStauch1 äquivalent, wenn es auf die korrekte Äußerung angewendet wird. Wie aus Abbildung 5.3 (links) ersehen werden kann, eignet sich diese Definition nicht so gut um Fehler aufzuzeigen, wenn es sich um eine estimierte Sprechgeschwindigkeit handelt.

5.2.2 Schwankung der estimierten Sprechgeschwindigkeit

Mit zwei weiteren Merkmalen wurde untersucht, ob das Auftreten von Schwankungen bei der estimierten Sprechgeschwindigkeit auf Erkennungsfehler hinweist.

Das erste Merkmal, das betrachtet wird, nennen wir Wortschwankung (WSchw). Besteht die Hypothese aus den N Worten w_1 bis w_N mit den dazugehörigen Sprechgeschwindigkeiten SpG_1 bis SpG_N, dann läßt sich für das Wort w_n WSchw mit Formel 5.1 berechnen. Zur Bestimmung der estimierten Sprechgeschwindigkeit wurde WStauch3 verwendet, wie in Abschnitt 5.2.1 beschrieben.

$$WSchw(w_n) := \begin{cases} \dfrac{SpG_n * 2}{SpG_{n-1} + SpG_{n+1}} & : \quad 1 < n < N \\ \dfrac{SpG_N}{SpG_{N-1}} & : \quad n = N \wedge N > 1 \\ \dfrac{SpG_1}{SpG_2} & : \quad n = 1 \wedge N > 1 \\ 1 & : \quad N = 1 \end{cases} \tag{5.1}$$

Durch den Vergleich mit dem Mittelwert, der aus dem Vorgänger- und Nachfolgewort berechnet wird, läßt sich eine sprechertypische Sprechgeschwindigkeit ausgleichen. Werden beispielsweise alle Worte mit einer (estimierten) Sprechgeschwindigkeit von 1,3 gesprochen, ergibt sich für jedes Wort ein Wert von 1. Wird dagegen ein einzelnes Wort deutlich schneller oder langsamer als die Umgebungsworte "gesprochen", wird diese Abweichung auch auf die Schwankung der Nachbarworte übertragen. Eine veränderte Sprechgeschwindigkeit deutet auf einen möglichen Erkennungfehler hin, der vielleicht die Erkennung des Nachfolgewortes negativ beeinflußt, oder selbst durch eine Fehlerkennung des Vorgängerwortes verursacht ist. Betrachtet man beispielsweise fünf Worte, bei denen die Geschwindigkeiten "1, 1, 2, 1, 1" festgestellt wurden, ergeben sich die Wortschwankung von "1, $\frac{2}{3}$, 2, $\frac{2}{3}$, 1". Die Beeinflussung der Schwankung der Nachbarworte ist deutlich zu sehen.

Beim zweiten Merkmal, das wir Kontextschwankung (KSchw) nennen, wird allein der Kontext um das betrachtete Wort untersucht. Die Berechnung erfolgt mit Formel 5.2, es gelten die gleichen Voraussetzungen wie für die Wortschwankung.

$$KSchw(w_n) := \begin{cases} \dfrac{SpG_{n-1}}{SpG_{n+1}} & : \quad 1 \leq n \leq N \\ 1 & : \quad n \in \{1, N\} \end{cases} \tag{5.2}$$

Es ist nur eine minimale Korrelation zur Wortschwankung gegeben, da das betrachtete Wort nicht in die Berechnung mit eingeht (Tabelle 5.12).

Beide Merkmale besitzen eine nur schwache Korrelation zum Gesamtfehler, die in Tabelle 5.11 angegeben ist. Es ist aber eine deutliche nichtlineare Beziehung vorhanden, wie aus Abbildung 5.4 zu ersehen ist. Anhand des Merkmals KSchw läßt sich erkennen, daß über den Kontext ebenfalls eine Beziehung zu Erkennungsfehlern besteht.

	Gesamt-FR	Einfüge-FR	OOV-FR
WSchw	2,3 %	1,3 %	1,0 %
KSchw	-0,5 %	-0,7 %	1,2 %

Tabelle 5.11: Korrelation: Wortschwankung und Kontextschwankung

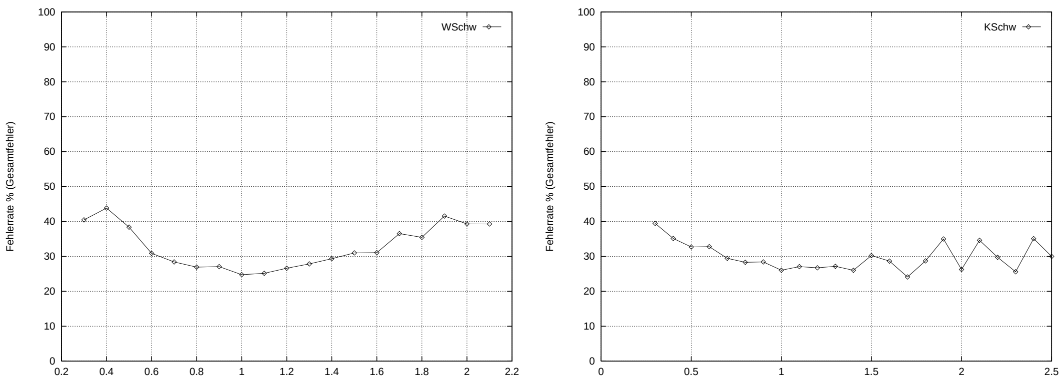


Abbildung 5.4: Gesamtfehler über die Schwankung der Sprechgeschwindigkeit (links WSchw, rechts KSchw)

5.2.3 Bewertung der zur Sprechgeschwindigkeit assoziierten Merkmale

Insgesamt läßt sich feststellen, daß zwischen der estimierten Sprechgeschwindigkeit und dem Auftreten von Erkennungsfehlern ein nichtlinearer Zusammenhang besteht. Positiv zu werten ist die Möglichkeit, statistisches Wissen über die akustische Trainingsmenge (Tabellen der Wortlängen) einzusetzen, um den Zusammenhang zwischen Erkennungsfehlern und Sprechgeschwindigkeit deutlicher hervortreten zu lassen.

Die Wortschwankung ist ein Merkmal, das nur den Kontext des Wortes verwendet. Da hier ein (geringer) Zusammenhang zur Fehlerrate erkennbar ist (Abbildung 5.4), läßt sich vermuten, daß ein Vertrauensmesser durch die Vergrößerung des Kontextes verbessert werden kann. Auf die Angabe einer zusammenfassenden Korrelationstabelle zu den Fehlerraten wird verzichtet, weil die Zusammenhänge im wesentlichen nichtlinear sind. Insgesamt kann die estimierte Sprechgeschwindigkeit als ein nützliches Merkmal für eine Klassifikation angesehen werden.

	WStauch3	WStreck3	WSchw	KSchw
WStauch3	100,0 %	-84,5 %	68,1 %	-3,1 %
WStreck3	-84,5 %	100,0 %	-57,7 %	-3,1 %
WSchw	68,1 %	-57,7 %	100,0 %	-0,6 %
KSchw	-3,1 %	-3,1 %	-0.6 %	100,0 %

Tabelle 5.12: Korrelation: Assoziierte Merkmale der Sprechgeschwindigkeit untereinander

5.3 Akustische Ähnlichkeit

Eine der am häufigsten untersuchte Eigenschaft der Worte einer Hypothese ist die Wort-Score, die bereits in Abschnitt 2.6 erklärt wurde. Als Äußerung wird eine Folge $A_{[1,T]} = \{a_1, ..., a_T\}$ von Merkmalsvektoren (Observationen) betrachtet. Die Wort-Score eines Worte w_n ist dabei durch einen Zeitausschnitt $T_{w_n} = [t_a^{w_n}, t_e^{w_n}]$ der Äußerung und der zum Wort w_n gehörenden Zustandsfolge $Z_{w_n} = \{z_{t_a}^{w_n}, z_{t_a+1}^{w_n}, ..., z_{t_e}^{w_n}\}$ (Viterbi-Pfad) des Modells λ_w definiert. Zustandsfolge und Zeitausschnitt werden durch den Viterbi-Algorithmus bestimmt. Die Zustandsfolge liegt auf dem global besten Pfad, da das Wort w_n aus der Hypothese stammt. Den Zuständen des Viterbi-Pfades ist jeweils ein Merkmalsvektor der Folge $A_{[t_a^w, t_e^w]}$ der Äußerung zugeordnet. Jeder Zustand steht für eine Wahrscheinlichkeitsverteilung, und ein Merkmalsvektor besitzt für einen gegebenen Zustand eine bestimmte Wahrscheinlichkeit, die wir als $z_t(a_t)$ schreiben. Die Wort-Score des Wortes w_n ergibt sich durch folgende Formel.

$$WScore(w_n) = -\log(\prod_{t=t_a}^{t_e} z_t(a_t)) \tag{5.3}$$

Mit der Wort-Score wurden folgende Merkmale berechnet und untersucht:

- mittlere Wort-Score (MWScore).
- mit einem Phonemerkenner normalisierte mittlere Wort-Score (PMWScore).
- mit der a priori Wahrscheinlichkeit eines kontextunabhängigen akustischen Modells normalisierte mittlere Wort-Score (AMWScore).
- Wort-Score dividiert mit der erwarteten Wortlänge (EMWScore).

Es ist zu bemerken, daß in allen Wort-Scores, die hier betrachtet wurden, die Wahrscheinlichkeit für das Sprachmodell enthalten ist.

5.3.1 Mittlere Wort-Score

Da die Wort-Score mit der Länge des gefundenen Wortes wächst, ist eine starke Korrelation mit der Wortdauer vorhanden. Um die unterschiedlich langen Zeitdauern der Worte auszugleichen, wird eine mittlere Wort-Score (MWScore) definiert, die sich nach folgender Formel berechnet.

$$MWScore(w_n) = \frac{WScore(w_n)}{t_e^w - t_a^w} \tag{5.4}$$

MWScore ist das geometrischen Mittel der Wahrscheinlichkeiten der Zustandsfolge $Z_w(A_{[t_a^w, t_e^w]})$, das noch logarithmiert und negiert wurde. MWScore ist somit ein Maß für die mittlere Ähnlichkeit der akustischen Merkmale mit dem Signal. Je kleiner dieser Wert ist, desto besser haben die akustischen Merkmale im Mittel auf das akustische Modell gepaßt. Es ist also anzunehmen, daß Worte mit einer kleinen MWScore häufiger richtig erkannt werden.

5.3.2 Viterbi-normalisierte Wort-Score

Würde bei der Berechnung der Hypothese auch die a priori Wahrscheinlichkeit $p(A)$ für das Auftreten einer Äußerung A in der Bayes-Formel (2.1) berücksichtigt, ergäbe sich für die Hypothese eine a posteriori Wahrscheinlichkeit $p(W|A)$, die bereits eine Bewertung der Glaubwürdigkeit darstellt. Es wird darauf hingewiesen, daß einerseits die Berechnung von $p(A)$ zu einem gegebenen Modell λ aufwendig ist, und andererseits die korrekte a posteriori Wahrscheinlichkeit nur mit einem korrekten Modell der gesprochenen Sprache bestimmt werden kann.[4]

Ward und Young [31][32] haben die Wort-Score untersucht und festgestellt, daß mit einer Approximation für $p(A)$, die mit einem Phonemerkenner gefunden wird, eine Verbesserung der Klassifikation von richtig und falsch erkannten Worten möglich ist. Dabei wurde eine Viterbi-Suche mit einem Spracherkenner und, parallel dazu, mit einem Phonemerkenner durchgeführt. Die Scores des Phonemerkenners wurde zum Schätzen des Terms $p(A)$ in der Bayes-Formel (2.1) verwendet, der zur Normalisierung dient. Da Scores logarithmierte Wahrscheinlichkeiten sind, muß für eine Normalisierung nur eine Subtraktion der Scores durchgeführt werden. Die normalisierte Wort-Score wurde abschließend auf eine einheitliche Wortdauer normiert. Die Berechnung erfolgt analog zu Formel 5.4.

5.3.3 A priori normalisierte Wort-Score

Cox und Rose [5] dagegen addieren für die Normalisierung die Wahrscheinlichkeiten der aktiven Zustände S_t in der Suche für jeden Zeitpunkt, wie es in folgender Formel dargestellt ist:

$$p(A_{[t_a,t_e]}) = \prod_{t=t_a}^{t_e} \frac{1}{S_t} \sum_{k=1}^{S_t} b_k(a_t) \tag{5.5}$$

Dabei ist S_t die Anzahl der aktiven Zustände in der Suche und $b_k(a_t)$ die Wahrscheinlichkeit des Merkmals a_t für den k-ten (aktiven) Zustand des Modells. Würde hier über alle Zustände des Modells addiert, ergäbe sich die a priori Wahrscheinlichkeit von $A_{[t_a,t_e]}$ für das betrachtete Modell. Eine Längenanpassung der normalisierten Wort-Score wurde nicht explizit durchgeführt. Die Dauer des Wortes ist aber ein Merkmal, das bei der Klassifikation verwendet wurde.

Die Approximation von $p(A)$ wurde in dieser Arbeit mit einem Phonemerkenner mit kontextunabhängigen akustischen Modellen durchgeführt. Diese Modelle wurden auf der gleichen akustischen Trainingsmenge bestimmt. Als Sprachmodell für die Phoneme wurde eine Gleichverteilung verwendet. Jedes Phonemmodell besteht aus sechs Zuständen, von denen mindestens drei durchlaufen werden müssen. Für die Berechnung der zum Normalisieren benötigten Scores, wurde für jede Äußerung eine Suche mit dem Phonemerkenner durchgeführt und anschließend der Viterbi-Pfad der besten Phonem-Hypothese gespeichert. Zu jedem Wort

[4] Es ist zu vermuten, daß trotz beachtlicher Erfolge in der Spracherkennung eine korrekte Modellierung noch nicht gefunden ist.

aus der Hypothese wurde darauf der Zeitausschnitt T_w ermittelt und der für die Normalisierung relevante Anteil des Viterbi-Pfades ausgewertet. Das Ergebnis ist eine Phonem-Score (PScore), die während des Zeitraumes des gefundenen Wortes im Phonemerkenner akkumuliert wurde (analog zur Formel 5.3). PMWScore berechnet sich dann mit folgender Formel

$$PMWScore(w_n) = \frac{WScore(w_n) - PScore(T_{w_n})}{t_e^{w_n} - t_a^{w_n}} \tag{5.6}$$

Die Differenz ist in diesem Fall eine Division von Wahrscheinlichkeiten ("Likelihood ratio"), da die Scores logarithmierte Wahrscheinlichkeiten approximieren.

Die gleichen kontextunabhängigen akustischen Modelle, wie sie im Phonemerkenner verwandt wurden, sind für die Berechnung der a priori Wahrscheinlichkeit eingesetzt worden. Sei S = $\{s_1, ..., s_N\}$ die Gesamtheit aller akustischen Modelle, $b_k(a_t)$ die Wahrscheinlichkeit, daß a_t im Zustand s_k auftritt, und A wie oben beschrieben, dann berechnet sich die akustische a priori Score (AScore) für den Zeitausschnitt $A_{[t_a,t_e]}$ von A mit folgender Formel

$$AScore(A_{[t_a,t_e]}) = -\log(\prod_{t=t_a}^{t_e} \frac{1}{N} \sum_{k=1}^{N} b_k(a_t)) \tag{5.7}$$

Die Normalisierung mit der a priori Score (AScore) erfolgt analog zur PScore in Formel 5.6. Als Ergebnis erhält man das Merkmal AMWScore.

Die Merkmale wurden daraufhin untersucht wie sie zu den Fehlern korrelieren. Das Ergebnis ist in Tabelle 5.13 zu sehen. Dabei ergibt sich, daß durch eine Normalisierung die Korrelation verbessert wird. Die Normalisierung mit der AScore liefert, abgesehen für Einfügefehler, bessere Werte.

5.3.4 Erwartete mittlere Wort-Score

Ein Merkmal, das ebenfalls aus einer Wort-Score berechnet wurde, soll hier noch erwähnt werden. Dabei wurde die Wort-Score nicht durch die Zeitdauer geteilt, die von der Suche gefunden wurde, sondern durch die erwartete Wortlänge (vgl. Abschnitt 5.13, Variante 3). Dieses Merkmal bezeichnen wir als eine 'erwartete' mittlere Wort-Score (EMWScore), sie ist das Produkt der Wortstreckung (Abschnitt 5.2.1) und der MWScore. Das Merkmal besitzt eine auffällig große Korrelation zu OOV-Fehlern. Diese beruht im wesentlichen auf der Tatsache, daß dieses Merkmal sehr gut mit Buchstabierungen korreliert (19,7 %).

Die Gesamtfehlerrate über dem Merkmal MWScore ist in Abbildung 5.5 zu sehen. Wie erwartet nimmt die Fehlerrate mit fallenden Werten ab. Überraschend

	Gesamt-FR	Einfüge-FR	OOV-FR
MWScore	12,4 %	7,6 %	2,6 %
PMWScore	13,1 %	9,5 %	3,9 %
AMWScore	13,5 %	8,8 %	5,9 %
EMWScore	10,7 %	8,5 %	11,8 %

Tabelle 5.13: Korrelation: (normalisierte) Wort-Score

ist aber das Ansteigen der Fehlerrate bei besonders niedrigen Scores, das durch das Ansteigen der Einfügefehlerrate verursacht wird. Dieser Effekt ist bei der AMWScore in Abbildung 5.6 (links) nicht so deutlich ausgeprägt.

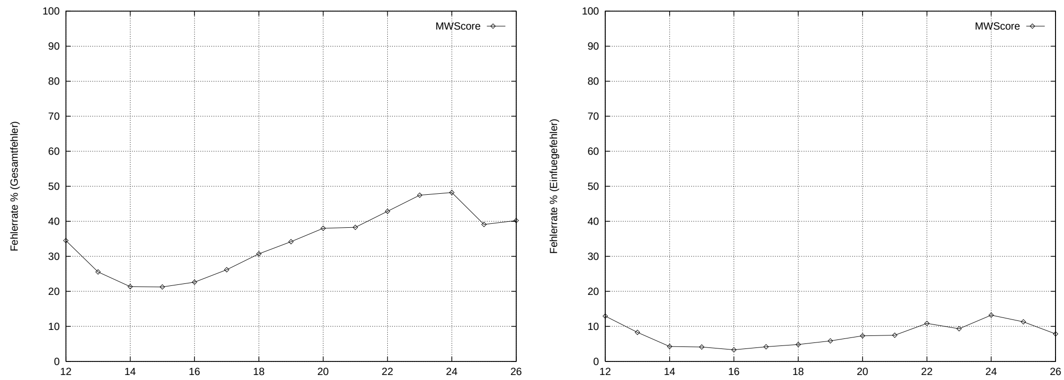


Abbildung 5.5: Gesamtfehler (links) und Einfügefehler (rechts) MWScore

5.3.5 Bewertung der akustischen Ähnlichkeit

Die mittlere akustische Ähnlichkeit eines Wortes (MWScore) besitzt erwartungsgemäß einen Zusammenhang zur Fehlerrate. Dieser Zusammenhang kann durch eine Normalisierung stärker hervorgehoben werden. Dabei ist die aufwendigere Normalisierung mit der a priori Wahrscheinlichkeit des kontextfreien akustischen Modells besser mit dem Fehler korreliert. Wie aus Tabelle 5.14 zu ersehen ist, sind beide Methoden der Normalisierung sehr eng miteinander korreliert. Eine Kombination der Merkmale in einem Klassifikator erscheint daher nicht sinnvoll. Das Produkt aus Wortstreckung und der mittleren akustischen Ähnlichkeit (EMWScore) ist dagegen mit den anderen Merkmalen vergleichsweise wenig, aber zu 88,4 % mit der Wortstreckung WStreck3 aus Abschnitt 5.2.1 korreliert. Das läßt vermuten, daß die Kombination der mit der Sprechgeschwindigkeit assoziierten Merkmale und einem Merkmal für akustische Ähnlichkeit in einem Klassifikator sinnvoll ist.

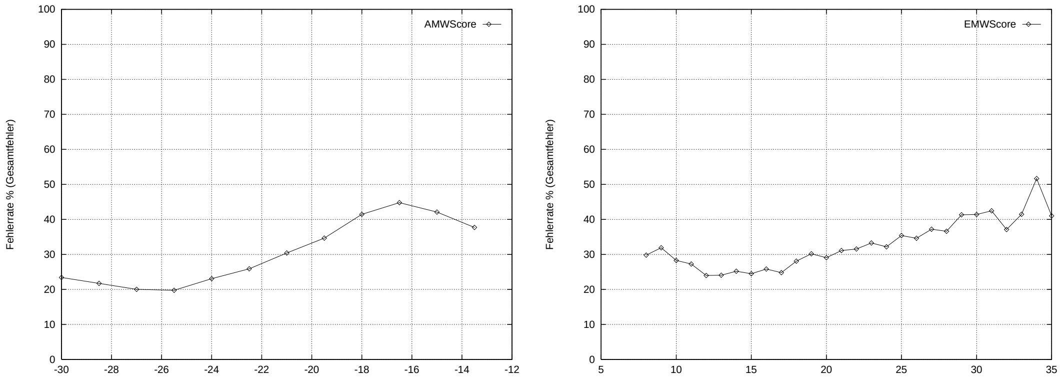


Abbildung 5.6: Gesamtfehler über AMWScore (links) und EMWScore (rechts)

	MWScore	PMWScore	AMWScore	EMWScore
MWScore	100,0 %	29,2 %	22,0 %	28,0 %
PMWScore	29,2 %	100,0 %	93,9 %	12,9 %
AMWScore	22,0 %	93,9 %	100,0 %	24,4 %
EMWScore	28,0 %	12,9 %	24,4 %	100,0 %

Tabelle 5.14: Korrelation: akustische Ähnlichkeit

5.4 Unsicherheit im Suchraum

Unterschiedliche Ursachen können einer Fehlerkennung zugrundeliegen, beispielsweise ein störendes Hintergrundgeräusch (Preßlufthammer), oder die Eingabe wird nicht durch ein akustisches Modell abgedeckt (beispielsweise ein unbekannter Name, Gestammel, etc.). Stimmen akustisches Modell und Sprachmodell gut mit einer beobachteten Äußerung überein, besitzt diese, im Vergleich zu alternativen Hypothesen, eine größere Wahrscheinlichkeit im Modell λ des Spracherkenners. Betrachtete man den Worthypothesengraph aus Abschitt 2.9 als eine kompakte Darstellung des Suchraumes, die alle möglichen Hypothesen enthält, so scheint der Viterbi-Pfad einer einfach und sicher zu erkennenden Äußerung wie ein breiter gut ausgewiesener Weg durch den Hypothesengraphen zu führen. Bei einer schwer zu erkennenden und leicht zu verwechselnden Äußerung dagegen, ist zu erwarten, daß sich viele ähnlich gute, beziehungsweise schlechte Pfade durch den Hypothesengraphen ergeben, und es somit viel einfacher ist, zufällig den falschen zu verfolgen. Es stellt sich nun die Frage, wie man eine Region erkennt, in der Unsicherheit besteht.

5.4.1 Akustische Stabilität

Cox, Rose [5] und Eide et al [7] untersuchten die N-Besten-Hypothesenliste, beziehungsweise den Worthypothesengraphen. Eide untersuchte, wie häufig ein betrachtetes Wort einer Hypothese in der N-Besten-Liste wiedergefunden werden kann. Je seltener es darin vorkomt, desto eher liegt ein Erkennungsfehler vor. Von den untersuchten Merkmalen erwies es sich als das aussagefähigste.

Qiu [21] berichtet von einem unveröffentlichten Artikel Zeppenfelds, bei dem ein ähnlicher Ansatz gewählt wurde. Es wurde aber nicht die N-Besten-Liste verwendet, sondern mit den zwei Parametern Sprachmodellgewicht (LG) und Wortübergangsstrafterm (WP) eine Hypothesenmatrix berechnet (vgl. Abschnitt 2.6). Finke [8], der zusammen mit Zeppenfeld dieses Merkmal entwickelt hat, nennt es akustische Stabilität (AStabil). Es wurde erfolgreich für die adaptive Sprecheranpassung mit glaubwürdigen Worten eingesetzt.

Die akustische Stabilität wurde für diese Arbeit als weiteres Merkmal untersucht. Zunächst wurde eine Matrix für Gewichtungsfaktoren des Sprachmodells und Strafterme für Wortübergänge festgelegt. Für eine Äußerung wurde der attribuierte Worthypothesengraph, und daraus die zu untersuchende Hypothese mit den Parametern LG_H und WP_H berechnet. Danach wurde zu jedem Parameterpaar (LG_A, WP_A) dieser Matrix aus dem Worthypothesengraphen eine alternative Hypothese berechnet (Hypothesenmatrix). Mit dem Align-Algorithmus (vgl. Abschnitt 3.1) wurde bestimmt, wie häufig ein betrachtetes Wort in der Hypothesenmatrix wiedergefunden werden konnte. Dieser Wert wurde abschließend durch die Anzahl der alternativen Hypothesen geteilt, wodurch sich Werte von 0.0 bis 1.0 ergeben können. Die Matrix mit den Werten für LG_A und WP_A kann dabei relativ zu dem Paar LG_H und WP_H festgelegt werden. Wurde beispielsweise die Hypothese mit dem Parameterpaar LG = 10 und WP = -3 erzeugt, können für die alternativen Hypothesen die beiden Parameter mit $\pm$ 1 , $\pm$ 2, $\pm$ 4, $\pm$ 8 und $\pm$ 16 modifiziert werden, daraus ergibt sich dann ein Matrix der Dimension 10×10. Bei der Beispielsmatrix kommt es sogar vor, daß der Sprachmodellgewichtungsfaktor LG negativ wird ($10 - 16 = -6$), was dazu führt, daß unwahrscheinlichere Wortfolgen durch das Sprachmodell bevorzugt werden. Ein gut passendes akustisches Modell steuert dem entgegen.

Das Merkmal wurde auf seine Korrelation zu den Fehlern untersucht. Abbildung 5.7 zeigt, daß für kleine Werte von AStabil ($< 0,4$) die Gesamtfehlerrate bis über 80 % steigt. Worte, deren akustische Stabilität in einem Bereich von 0.975 bis 1.0 liegt, werden dagegen zu 91,9 % richtig erkannt.

Es besteht eine außerordentlich deutliche, Korrelation zur Fehlerrate, wie aus Tabelle 5.15 ersichtlich ist. Mit diesem Merkmal läßt sich auf elegante und einfache Weise feststellen, wo unsichere Bereiche in einer Hypothese sind, die zu einem Erkennungsfehler führen können. Ein wesentlicher Vorteil von AStabil ist,

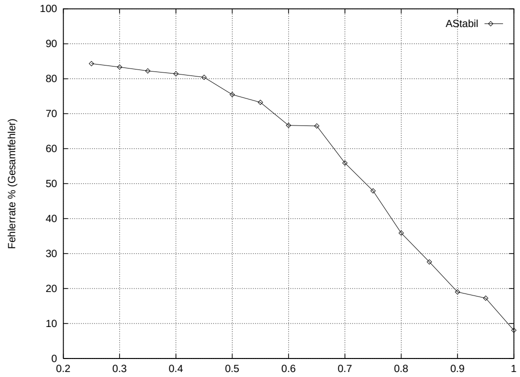


Abbildung 5.7: Gesamtfehler über akustischer Stabilität (AStabil)

daß die Berechnung ohne die Vorgabe einer zeitlichen Segmentierung der Äußerung geschieht, die eventuell beschränkend wirken könnte.

	Gesamt-FR	Einfüge-FR	OOV-FR
AStabil	-54,0 %	-20,6 %	-18,1 %

Tabelle 5.15: Korrelation: akustische Stabilität

5.4.2 Entropie im Worthypothesengraph

Shannon [30] erklärt, daß Wahlfreiheit, beziehungsweise Verwechselbarkeit ('äquivokation' nach Shannon), sich als Entropie ausdrücken läßt (vgl. Abschnitt 2.10).

Ein Hypothesengraph, dessen Kanten mit Übergangswahrscheinlichkeiten attribuiert sind, stellt, im Sinne der Informationstheorie, eine Nachrichtenquelle für Hypothesen dar. Die Worte der Hypothese können mit einer gewissen Wahlfreiheit, beziehungsweise Verwechselbarkeit, erzeugt werden, die durch die Übergangswahrscheinlichkeiten im Hypothesengraphen festgelegt ist.

Nachdem nun ein Hypothesengraph als Nachrichtenquelle aufgefaßt werden kann, analog zu einem Markov-Modell, kann die Entropie für den gesamten Worthypothesengraphen berechnet werden. Da aber nicht die Hypothese als Ganzes, sondern nur einzelne Worte daraus bewertet werden sollen, muß die Berechnung der Entropie auf einen Zeitausschnitt der Äußerung beschränkt werden. Dabei ergeben sich verschiedene Schwierigkeiten, die besonders mit Kanten zusammenhängen, die über die gewählten Zeitgrenzen hinausreichen. Anhang A bietet für dieses Problem verschiedene Lösungsansätze an und beschreibt die konkrete Berechnung. Für die hier durchgeführte Berechnung wurden Wahrscheinlichkeiten im Worthypothesengraphen verwendet, die mit dem Viterbi-Algorithmus gefunden wurden. Diese nähern nur die tatsächlichen Übergangswahrscheinlichkeiten an. Probleme, die sich daraus ergeben können, werden ebenfalls im Anhang näher beschrieben.

Für die Berechnung der Entropie wurde die Datenstruktur des Worthypothesengraphen (Lattice) erweitert, damit verschiedene Berechnungsparameter effizient untersucht werden konnten. Die Implementierung besitzt folgende Parameter:

- Anfangszeitpunkt/Endzeitpunkt: Start- beziehungsweise Endpunkt des Zeitintervalls, in dem die Knoten liegen, die für die Berechnung der Entropie herangezogen werden. Normalerweise der Zeitpunkt, an dem das untersuchte Wort beginnt, beziehungsweise endet. Soll nur die Unsicherheit zu Beginn oder am Ende des Wortes bestimmt werden, setzt man beide Werte entweder auf den Wortanfang oder auf das Wortende. Es werden dann nur die Kanten betrachtet, die zu diesem Zeitpunkt von den Knoten abgehen. Bei der gewählten Implementierung werden abgehende Kanten von Knoten innerhalb des gewählten Zeitintervalls immer in die Berechnung mit einbezogen.

- Richtung: Orientierung der Kanten mit oder gegen die Zeitrichtung des Graphen. Damit ist ebenfalls festgelegt, ob die "Startknoten" für die Berechnung zu Beginn oder am Ende des Zeitintervalls liegen.

- Delta: Bei der Implementierung wurde ein virtueller Knoten eingeführt, der die "Startknoten" zusammenfaßt. Ist Delta $= 0$ werden nur die Knoten, die exakt auf dem Startzeitpunkt liegen, mit diesem Knoten verbunden. Für ein Delta > 0 werden alle Knoten, die höchstens $\pm$ Delta von diesem Zeitpunkt entfernt liegen, mit dem virtuellen Knoten verbunden. Eine fest vorgegebene zeitliche Segmentierung kann Probleme bereiten, die durch das Delta gemildert werden.

- Sprachmodell: Berechnung unter Einbeziehung der Wahrscheinlichkeiten für Wortübergänge, die aus dem Sprachmodell stammen, beziehungsweise ohne diese.

- Längennormierung: Normalerweise wird die Wahrscheinlichkeit einer Kante des Worthypothesengraphen ohne Längennormierung verwendet. Eine Normierung der Länge wurde experimentell aufgenommen. Dabei wird das geometrische Mittel der Wahrscheinlichkeiten des Viterbi-Pfades, der einer Kante entspricht, als Übergangswahrscheinlichkeit für eine Kante des Graphen verwendet (analog zur mittleren Wort-Score (MWScore) aus Abschnitt 5.3).

Durch den Einsatz der Technik des dynamischen Programmierens war es möglich 168 Parameterkombinationen zu untersuchen. Dabei zeigte sich, daß die in Tabelle 5.16 dargestellte Parameterkombinationen, die wir H1, H2 und H3 nennen, besonders gut mit den Fehlerraten korreliert sind. Die Korrelation zu den Fehlern ist aus Tabelle 5.17 zu ersehen.

Parameter	H1	H2	H3
Startzeitpunkt	Wortanfang	Wortanfang	Wortanfang
Endzeitpunkt	Wortende	Wortanfang	Wortende
Richtung	mit Zeit	gegen Zeit	gegen Zeit
Delta	$\pm$ 25 ms	$\pm$ 75 ms	$\pm$ 55 ms
Sprachmodell	Einbeziehung	Einbeziehung	Einbeziehung
Längennormierung	nein	nein	ja

Tabelle 5.16: Parameterkombinationen mit deutlicher Korrelation zur Fehlerrate

	Gesamt-FR	Einfüge-FR	OOV-FR
H1	47,9 %	18,4 %	16,0 %
H2	46,9 %	18,6 %	16,5 %
H3	35,5 %	11,9 %	19,5 %

Tabelle 5.17: Korrelation: Entropie

Bei der Untersuchung erwies sich, daß das Einbeziehen des Sprachmodells zu einer besseren Korrelation mit der Fehlerrate führt. War es nicht berücksichtigt, wurde im allgemeinen eine gute Korrelation mit größeren Werten für den Parameter Delta erreicht. Bemerkenswert ist, daß sich für ein Delta von $\pm$ 25 ms (H1) die deutlichste Korrelation ergab. Es besteht möglicherweise ein Zusammenhang zur durchschnittlichen Phonemlänge, die im Deutschen ungefähr 50 ms beträgt. Merkmal H2 besitzt eine vergleichbar gute Korrelation zum Gesamtfehler, ist aber von den 168 untersuchten Parameterkombinationen am besten mit den Einfügefehlern korreliert. Bemerkenswert ist, daß hier die Entropie nicht von Wortanfang bis Wortende berechnet wurde, sondern nur für den Wortanfang. Durch das große Delta ergibt sich aber eine Zeitspanne, die viele kurze Worte erfaßt. Unerwarteterweise ist ein Merkmal, das die (experimentelle) Längennormierung verwendet, von allen untersuchten Merkmalen am besten mit den OOV-Fehlern korreliert (H3).

Die Gesamtfehlerrate über der Entropie ist in Abbildung 5.8 aufgetragen. Es ist zu erkennen, daß eine im wesentlichen lineare Beziehung besteht. Für Merkmal

H3 wächst die Fehlerrate nicht so deutlich wie für die Merkmale H1 und H2. Für die Merkmale H1 und H2 steigt ab ungefähr 4,5 Bit die Fehlerrate auf über 50 % an. Gibt es also im Mittel mehr als 23 gleich gute Möglichkeiten durch den Hypothesengraphen zu gelangen liegt häufiger ein Erkennungsfehler anstatt einer korrekten Erkennung des Wortes vor. Nahe 0 Bit werden bei allen drei Merkmalen sehr niedrige Fehlerraten erreicht (bei Merkmal H1 3,9 %). Fehler die hier auftreten, sind schwer zu entdecken und bestehen, wie aus Abbildung 5.10 zu erkennen ist, zu einem großen Teil aus OOV-Fehlern (bei Merkmal H1 1,7 %). OOV-Worte können zu Fehlerkennungen führen, die gut auf die akustischen Modelle und Sprachmodelle passen (vgl. «füreinander» in Abschnitt 3.1). Bei Merkmal H1 beträgt die Einfügefehlerrate 1,1 % in der Nähe von 0 Bit (Abbildung 5.9). Insgesamt bestehen beim Merkmal H1 die verbliebenen Fehler nahe einer Entropie von 0 Bit zu über 70 % aus OOV-Fehlern und Einfügefehlern.

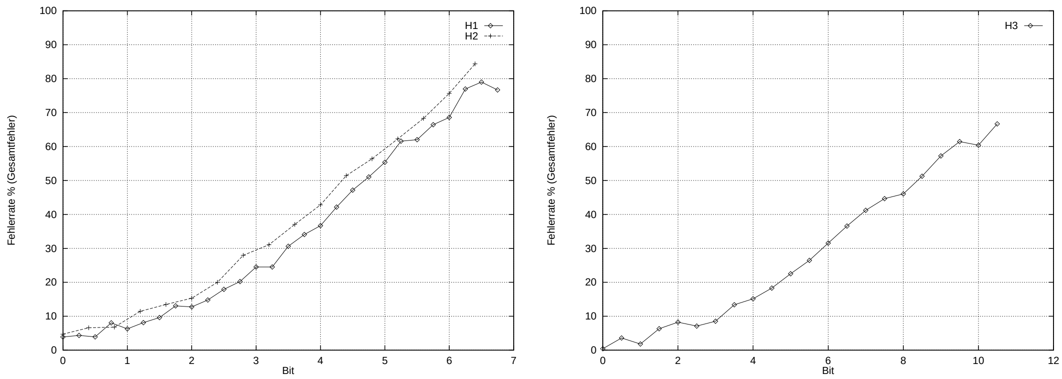


Abbildung 5.8: Gesamtfehler über H1, H2 (links) und H3 (rechts)

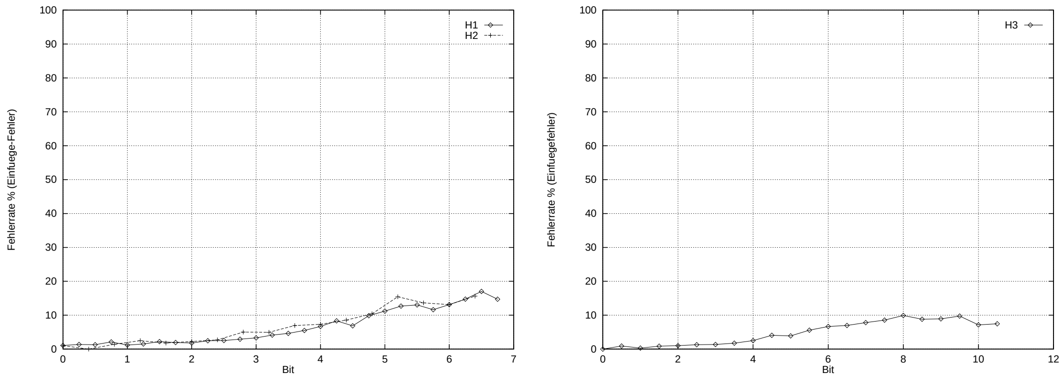


Abbildung 5.9: Einfügefehler über H1, H2 (links) und H3 (rechts)

OOV-Fehler werden aber (Abbildung 5.10) gut durch Merkmal H3 ausgeschlossen. Nahe einer Entropie von 0 Bit ist die OOV-Fehlerrate mit ungefähr 0,4 % sehr niedrig. Es ist anzumerken, daß die Entropie des Merkmals H3 wegen der wesentlich anderen Berechungsvorschrift nicht direkt mit den Merkmalen

H1 und H2 verglichen werden kann. Ein möglicher Effekt der Längennormierung könnte sein, daß das Sprachmodell, das den Worthypothesengraphen strukturiert hat, stärker die Berechnung der Entropie beeinflußt, und somit OOV-Fehler verstärkt hervorgehoben werden.

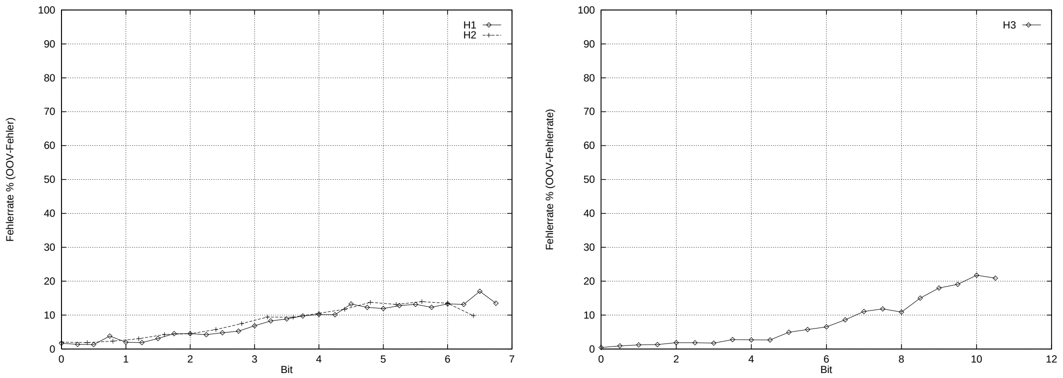


Abbildung 5.10: OOV-Fehler über H1, H2 (links) und H3 (rechts)

5.4.3 Bewertung der Unsicherheit im Suchraum

Die Merkmale, die die Unsicherheit im Suchraum messen, zeigen eine besonders gute Korrelation zu den Fehlerraten. Eine Unsicherheit, die zu Erkennungsfehlern führt, kann dabei völlig unterschiedliche Gründe haben, die alle im Worthypothesengraphen zusammenfließen. Die akustische Stabilität ist dabei zu -54,0 % mit dem Gesamtfehler korreliert. Diese Korrelation läßt sich möglicherweise durch eine andere Parametermatrix verbessern. Eine besonders hervorzuhebende Eigenschaft dieses Merkmals ist, daß es keine festgelegten Wortgrenzen benötigt.

Auch die Merkmale für die Entropie in der Suche sind möglicherweise noch zu verbessern, indem die Übergangswahrscheinlichkeiten im Worthypothesengraphen nicht mit dem Viterbi-Algorithmus approximiert, sondern mit dem Forward-Backward-Algorithmus [22] bestimmt werden. Die akustische Stabilität und die Entropiemaße sind leider eng miteinander korreliert (vgl. Tabelle 5.18), so daß eine Verbesserung eines Vertrauensmessers durch Kombination der Merkmale unwahrscheinlich erscheint.

Der Grund für die hervorragende Korrelation zwischen den OOV-Fehlern und dem experimentellen Merkmal H3, bei dem eine Längennormierung stattfindet, ist unklar. Möglicherweise liegt die Ursache im stärkeren Einfluß des Sprachmodells, das den Worthypothesengraphen strukturiert hat. Als Merkmal für eine Bewertung der Glaubwürdigkeit einer Hypothese ist diese deutliche Korrelation aber sehr nützlich, zumal Merkmal H3 verhältnismäßig schwach mit den anderen Merkmalen korreliert ist. Erwartungsgemäß sind die Merkmale H1 und H2 eng

	AStabil	H1	H2	H3
AStabil	100,0 %	-66,1 %	-65,7 %	-40,1 %
H1	-66,1 %	100,0 %	91,7 %	46,5 %
H2	-65,7 %	91,7 %	100,0 %	45,6 %
H3	-40,1 %	46,5 %	45,6 %	100,0 %

Tabelle 5.18: Korrelation: Unsicherheit im Suchraum

miteinander korreliert. Merkmal H2 ist aber etwas besser mit Einfügefehlern korreliert, und somit weiterhin für einen Klassifikator interessant. Ein beachtenswerter Aspekt der Entropieberechnung ist, daß eine sehr effiziente Implementierung möglich ist, die das Merkmal auch für den Einsatz in echtzeitfähigen Spracherkennern geeignet erscheinen läßt.

Da die untersuchten Merkmale sehr deutlich linear mit den Fehlern korrelieren ist zu erwarten, daß bereits mit verhältnismäßig einfachen Klassifikatoren gute Erfolge erzielt werden können. Beispielsweise könnte ein Neuronales Netz mit nur einer versteckten Einheit und direkten Verbindungen zwischen Eingabe- und Ausgabeneuronen eingesetzt werden. Die Auswertung kann dann ebenfalls sehr schnell durchgeführt werden, was sich für echtzeitfähige Spracherkenner wiederum sehr positiv bemerkbar macht.

Kapitel 6

Experimente und Ergebnisse

In diesem Kapitel wird eine Teilmenge der vorgestellten Merkmale daraufhin untersucht, wie gut sie die Bewertungsfähigkeit eines Vertrauensmessers verbessern können. Da es Anzeichen gibt, die darauf hindeuten, daß eine Vergrößerung des Kontextes die Klassifikationleistung verbessern könnte, wird dies ebenfalls untersucht. Als zu optimierendes Qualitätsmaß wurde das NIST-Qualitätsmaß (NIST-QM) gewählt. Das binäre Qualitätsmaß (B-QM) ist interessehalber ebenfalls angegeben, die Ergebnisse sind aber nicht notwendigerweise optimal.

6.1 Der Klassifikator

Für die Klassifikation wurden Neuronale Netze trainiert, die einen besonders einfachen Aufbau haben. Zwischen den Eingabeeinheiten und den Ausgabeeinheiten existieren direkte Verbindungen ("Shortcuts"); es wird nur eine einzige versteckte Einheit verwendet. Abbildung 6.1 zeigt eine solche Netzstruktur für beispielsweise zwei Eingabemerkmale. Ein Neuronales Netz, das solch eine Struktur aufweist, kann bereits das XOR-Problem lösen [28] und ist somit *kein* linearer Klassifikator mehr.

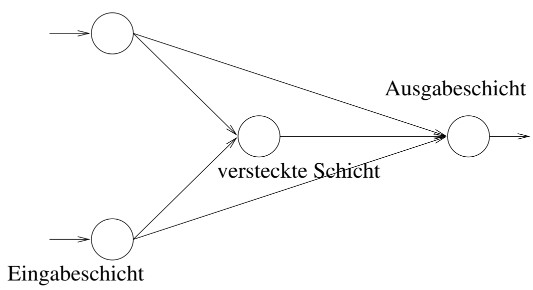


Abbildung 6.1: Eingesetzte Netzstruktur

Für jede Untersuchung wurden drei Netze mit dieser Grundstruktur trainiert und auf einer unabhängigen Kreuzvalidierungsstichprobe getestet. Das Netz, das auf der Kreuzvalidierungsstichprobe das beste Ergebnis lieferte, wurde für die Evaluation eingesetzt. Für das Training wurde der Stuttgarter Neuronale Netze Simulator (SNNS) in der Version 4.1 mit dem Standardverfahren für Fehlerrückpropagierung (StdBackprop) verwendet. Die zu optimierende Fehlerfunktion war dabei, bedingt durch die Verwendung des SNNS, die Fehlerquadratesumme. Für das NIST-QM wäre die "Cross-Entropy" die geeignetere Fehlerfunktion, da das NIST-QM sich mit dieser Funktion ausdrücken läßt (vgl. Abschnitt 3.3) und somit das gewünschte Qualitätsmaß direkt optimiert würde.

6.2 Merkmalkombinationen

Als Trainingsmenge wurden die in Abschnitt 4.2 beschriebenen 14906 Hypothesenworte verwendet, für die Auswahl des besten Netzes diente die Kreuzvalidierungsstichprobe mit 9002 Worten. Die Klassifikationsleistung des Netzes wurde dann auf den 3839 Worten der Evaluationsmenge (evalset-short) gemessen. Zur Erinnerung sei nochmals erwähnt, daß diese Menge, im Gegensatz zur Kreuzvalidierungsstichprobe und zur Trainingsmenge, *keine* Buchstabierungen enthält.

Die für die Klassifikation ausgewählten Merkmale sind (vgl. Kapitel 5):

- Unsicherheit im Suchraum : AStabil, H1, H2, H3
- Sprechgeschwindigkeit : WStreck3, WStauch3
- Wortassoziierte Merkmale : SM-NGRAM, LogPhonAnz, LogAnzTrain
- Akustische Ähnlichkeit : MWScore, EMWScore

Die Verwendung der nicht normalisierten mittleren Wort-Score (MWScore) bei der akustischen Ähnlichkeit hat zwei Gründe. Zum einen ist die mittlere Wort-Score besonders einfach zu erhalten, und somit für den Einsatz in echtzeitfähigen Spracherkennern besonders interessant, zum anderen stand der kontextunabhängige Spracherkenner für die Normalisierung während der Auswertung nicht zur Verfügung.

In Tabelle 6.1 sind die Korrelationen der ausgewählten Merkmale zu den Fehlern zusammengefaßt. Die Korrelation der Merkmale untereinander ist in den Tabellen 6.2 und 6.3 angegeben.

Für die Untersuchung des Kontexteinflusses in der Klassifikationsleistung wurden die Nachbarworte herangezogen und der Merkmalsvektor des untersuchten

	Gesamt-FR	Einfüge-FR	OOV-FR
AStabil	-54,0 %	-20,6 %	-18,1 %
H1	47,9 %	18,4 %	16,0 %
H2	46,9 %	18,6 %	16,5 %
H3	35,5 %	11,9 %	19,5 %
WStauch3	-1,3 %	-1,6 %	-8,2 %
WStreck3	6,6 %	6,0 %	11,5 %
LogPhonAnz	-11,8 %	-12,1 %	-1,0 %
SM-NGRAM	-19,6 %	-0,7 %	-14,3 %
LogAnzATrain	-10,7 %	-0,3 %	-16,2 %
MWScore	12,4 %	7,6 %	2,6 %
EMWScore	10,7 %	8,5 %	11,8 %

Tabelle 6.1: Korrelation: Ausgewählte Merkmale

Wortes um die Merkmale der Nachbarworte erweitert. Dabei wurden für alle Worte die gleichen Merkmale eingesetzt. Beispielsweise AStabil für das linke Nachbarwort, das betrachtete Wort und für das rechte Nachbarwort. Falls dies nicht möglich war, weil beispielsweise das betrachtete Wort das erste der Hypothese war, wurden der Einfachheit halber die Werte des betrachteten Wortes für das fehlende Nachbarwort eingesetzt. Die Werte stammen nur von zu klassifizierenden Worten, da Müllworte übersprungen werden. Beispielsweise hat das Wort ≪also≫ in ≪#atmen# gut #ähm# also ja≫ als linken Nachbarn ≪gut≫ und als rechten ≪ja≫. Das Wort ≪gut≫ besitzt dagegen keinen linken Nachbarn. Tabelle 6.4 zeigt die untersuchten Merkmalskombinationen M01 bis M12, die für die Klassifikatoren herangezogen wurden.

6.3 Ergebnisse

Tabelle 6.5 zeigt die Klassifikationsleistung der gefundenen Netze auf der Evaluationsmenge. Zusätzlich ist die Klassifikationsleistung auf der Kreuzvalidierungsstichprobe (Kreuzval.) angegeben, da dies einen Rückschluß über die Generalisierungsfähigkeit erlaubt. In der Tabelle steht "oK" für "ohne Kontext" und "mK" für "mit Kontext".

Die Merkmalskombinationen M01, M02 und M03 wurden gewählt, um festzustellen, wie geeignet die Merkmale akustische Stabilität und die Entropie im Worthypothesengraphen für sich alleine betrachtet sind. M03 wurde in diesem Sinne als ein Merkmal angesehen, das mit drei Kennzahlen die Unsicherheit auf

	AStabil	H1	H2	H3	WStauch3
H1	-66,1 %	100,0 %			
H2	-65,7 %	91,7 %	100,0 %		
H3	-40,1 %	46,5 %	45,6 %	100,0 %	
WStauch3	-12,0 %	20,1 %	21,5 %	-12,0 %	100,0 %
WStreck3	6,5 %	-13,7 %	-15,2 %	13,0 %	-84,0 %
LogPhonAnz	10,3 %	-20,5 %	-22,6 %	0,5 %	-0,4 %
SM-NGRAM	20,6 %	-22,6 %	-22,5 %	-26,4 %	0,5 %
LogAnzATrain	11,5 %	-1,2 %	-1,8 %	-15,0 %	6,2 %
MWScore	-15,8 %	19,3 %	19,0 %	7,1 %	19,1 %
EMWScore	0,4 %	-6,0 %	-7,7 %	15,4 %	-74,3 %

Tabelle 6.2: Korrelation: Ausgewählte Merkmale untereinander (Teil 1)

	WStreck3	LogPhon-Anz	SM-NGRAM	LogAnz-ATrain	MWScore
LogPhonAnz	-10,0 %	100,0 %			
SM-NGRAM	-4,6 %	-10,7 %	100,0 %		
LogAnzATrain	0,4 %	-54,5 %	29,2 %	100,0 %	
MWScore	-14,8 %	-14,6 %	3,1 %	10,9 %	100,0 %
EMWScore	88,4 %	-14,6 %	-3,5 %	4,1 %	28,3 %

Tabelle 6.3: Korrelation: Ausgewählte Merkmale untereinander (Teil 2)

Basis der Entropie mißt. Ohne Kontext ist dann auch die Merkmalskombination M03 mit einem NIST-QM von 0,204 besser als die Merkmalskombinationen M01 (0,199) und M02 (0,191). Durch Hinzunahme des Kontextes verbesserten sich alle drei Merkmalskombinationen. Die akustische Stabilität gewann dabei am meisten und ist mit einem NIST-QM von 0,218 besser als die Entropie im Worthypothesengraphen M02 (0,201) und M03 (0,217).

Da die akustische Stabilität und die Entropie im Worthypothesengraphen teilweise sehr deutlich miteinander korreliert sind (vgl. Tabelle 6.1), war eine Verbesserung durch eine Kombination der Merkmale nicht unbedingt zu erwarten. Die Merkmalskombination M04 belegt, daß eine Verbesserung aber trotzdem möglich ist. Es kann im wesentlichen zwei Gründe haben, daß der Gewinn so deutlich ausfällt. Entweder messen beide Merkmale unterschiedliche Eigenschaften oder die Parameter der Merkmale sind nicht optimal bestimmt.

Die Hinzunahme der mit der Sprechgeschwindigkeit assoziierten Merkmale WStreck3 und WStauch3 (M05) ergab noch einmal eine deutliche Verbesserung des NIST-QM. Der Grund ist möglicherweise, daß Buchstabierungen in der Trai-

Merkmals-kombination	Merkmale
M01	AStabil
M02	H1
M03	H1, H2, H3
M04	AStabil, H1, H2, H3
M05	M04 + WStreck3, WStauch3
M06	M05 + LogPhonAnz, LogAnzATrain, SM-NGRAM
M07	M06 + MWScore, EMWScore
M08	M06 + MWScore
M09	M06 + EMWScore
M10	WStreck3, WStauch3, LogPhonAnz, LogAnzATrain, SM-NGRAM
M11	M08 + MWScore
M12	M08 + EMWScore

Tabelle 6.4: Untersuchte Merkmalskombinationen

ningsmenge besser getrennt werden konnten, und somit die Evaluationsmenge besser geschätzt werden kann. Ein weiterer Grund kann sein, daß Fehlerkennungen auch untypische Wortlängen produzieren, die weder bei der Berechnung der akustischen Stabilität noch der Berechnung der Entropie im Worthypothesengraphen berücksichtigt sind.

Durch Hinzunahme der wortassoziierten Merkmale (M06) ließ sich noch einmal eine Verbesserung erreichen. Wie erwartet führt ein größerer Kontext zu besseren Ergebnissen. Das NIST-QM ist mit 0,291 dabei das beste Ergebnis in Tabelle 6.5. Mit 80 % beim B-QM wurde ebenfalls das beste Ergebnis mit dieser Merkmalskombination erreicht; das entspricht einer Fehlerreduktion von 36,9 % (Tabelle 6.6).

Überraschend ist, daß durch Hinzunahme der Merkmale für die akustische Ähnlichkeit die Klassifikationsleistung insgesamt zurückfällt. Das Ergebnis überrascht insbesondere deshalb, da auf der Kreuzvalidierungsstichprobe eine Verbesserung des Nist-QM erreicht wurde.

Um festzustellen, welches der beiden Merkmale aus der Gruppe der akustischen Ähnlichkeit hierfür verantwortlich sein könnte, wurden die Merkmalskombinationen M08 und M09 untersucht. Als Ursache für die Verschlechterung wird einerseits die Inhomogenität der Daten angesehen, da ungefähr gleich gute Ergebnisse auf der Kreuzvalidierungsstichprobe erreicht werden konnten, andererseits scheint gerade die Kombination der beiden Merkmale die Klassifikationsfähigkeit zu beeinträchtigen. Jedes Merkmal für sich auf der Kreuzvalidierungsstichprobe

	Evaluation				Kreuzvalid.			
	NIST-QM		B-QM		NIST-QM		B-QM	
	oK	mK	oK	mK	oK	mK	oK	mK
A Priori	0.000	0.000	68,3 %	68,3 %	0.000	0.000	73,8 %	73,8 %
M01	0,199	0,218	77,0 %	77,1 %	0,240	0,260	81,5 %	81,4 %
M02	0,190	0,201	75,6 %	76,0 %	0,209	0,229	78,8 %	79,2 %
M03	0,204	0,217	76,2 %	76,4 %	0,241	0,255	79,8 %	80,0 %
M04	0,244	0,253	77,5 %	77,5 %	0,293	0,303	81,9 %	82,1 %
M05	0,274	0,279	78,4 %	78,7 %	0,309	0,319	82,3 %	82,9 %
M06	0,282	**0,291**	79,3 %	**80,0** %	0,320	0,337	82,9 %	83,6 %
M07	0,273	0,283	79,2 %	79,1 %	0,325	0,339	83,0 %	83,6 %
M08	0,288	0,282	78,9 %	79,1 %	0,330	0,339	83,2 %	83,9 %
M09	0,280	0,289	78,7 %	79,6 %	0,328	0,341	83,2 %	83,7 %
M10	0,067	0,082	70,9 %	72,2 %	0,090	0,138	76,9 %	77,2 %
M11	0,050	0,082	70,5 %	71,6 %	0,105	0,155	77,1 %	78,0 %
M12	0,060	0,087	70,0 %	71,6 %	0,102	0,148	76,8 %	77,6 %

Tabelle 6.5: Klassifikationsergebnisse

und auf der Evaluationsmenge genommen erreicht entweder mit oder ohne Kontext gute Werte. Daß die Kombination M09 bei Verwendung des Kontextes besser ist, kann daran liegen, daß Buchstabierungen besser erkannt werden. EMWScore ist gut mit OOV-Worten korreliert. Ohne Kontext ist aber MWScore besser.

Nachdem die Merkmalskombination M06 die besten Ergebnisse erzielt hatte, wurde untersucht wie gut die Kombination ohne die Merkmale der Unsicherheit in der Suche ist. Dazu wurden die Merkmalskombinationen M10, M11 und M12 betrachtet. Bei diesen Kombinationen ist ebenfalls der Kontext von Vorteil. Mit einem NIST-QM von 0,087 (M12) ist wird aber deutlich, daß ohne die Merkmale akustische Stabilität oder Entropie im Worthypothesengraphen mit den verbleibenden Merkmalen keine gute Vertrauensmessung möglich ist.

	Fehlerreduktion
a priori	0 %
Merkmale mit Kontext: AStabil, H1, H2, H3, WStreck3, WStauch3, LogPhonAnz, LogAnzATrain, SM-NGRAM	36,9 %

Tabelle 6.6: Fehlerreduktion des besten Klassifikators

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde aufgezeigt, daß es viele Anwendungsmöglichkeiten für einen Vertrauensmesser in der maschinellen Spracherkennung gibt, beispielsweise ein verbesserter Mensch-Maschine-Dialog und die Verbesserung der Erkennungsleistung von Spracherkennern durch Unterstützung bei der Sprecheradaption.

Die Bedeutung der Entropie als Maß für Unsicherheit wurde dargelegt und erläutert, wie die Entropie eines Graphen berechnet werden kann.

Anschließend wurden der verwendete Spracherkenner Janus-3 und die verwendete Datenbasis beschrieben. Dabei wurde darauf hingewiesen, daß die Trainingsmenge und die Evaluationsmenge Inhomogenität aufweisen, da die Trainingsmenge Buchstabierungen enthält, die mit dem verwendeten System nicht erkannt werden können.

Es wurde erklärt, wie mit dem Align-Algorithmus eine Einteilung in Fehlerarten (Ersetzungsfehler, Einfügefehler und OOV-Fehler) möglich ist und dargelegt, welche Schwierigkeiten sich dabei ergeben. Um objektive Aussagen über die Qualität eines Vertrauensmessers und dessen Verbesserung machen zu können, werden Qualitätsmaße benötigt. Verschiedene Maße wurden betrachtet und bewertet. Genauer untersucht wurde die "normalisierte Cross-Entropy" (NIST-QM). Dieses Maß erscheint besonders interessant, da die Qualität die a posteriori Wahrscheinlichkeit $p(Korrekt|w)$ eines Wortes w zu schätzen gemessen wird.

Für die Konstruktion eines Vertrauensmessers ist es notwendig, eine Menge von aussagefähigen Merkmalen zu finden. Verschiedene Merkmalsgruppen wurden untersucht und in den Zusammenhang aktueller Forschungen gestellt. Als Bewertungsmaß ob ein Merkmal für einen Vertrauensmesser in Frage kommt diente zuerst die Korrelation mit den Fehlerarten. Zusätzlich wurde eine graphische Darstellung der Fehlerrate über dem Merkmal verwendet, da eine hohe Korrelation zwar ein gutes Merkmal andeutet, eine niedrige aber nicht auf die Unbrauchbarkeit des Merkmals schließen läßt.

Folgende Gruppen von Merkmalen wurden untersucht:

1. Wortassoziierte Merkmale: Wortidentität, Wortlänge, Sprachmodell, Häufigkeit im akustischen Training.

2. Sprechgeschwindigkeit: Wortstreckung, Wortstauchung.

3. Akustische Ähnlichkeit: mittlere Wort-Score (MWScore), normalisierte MWScore.

4. Unsicherheit im Suchraum: akustische Stabilität, Entropie im Worthypothesengraphen.

Bei den wortassoziierten Merkmalen zeigte sich, daß sich die Fehlerraten einzelner Worte der Hypothese stark unterscheiden, weshalb die Wortidentität als ein aussichtsreiches Merkmal erscheint. Eine Abhängigkeit der Fehlerrate von der Wortlänge konnte bestätigt werden. Es zeigte sich, daß eine logarithmische Längenmessung die Korrelation zu den Fehlern verbessert. Das Sprachmodell liefert ein gut korreliertes Merkmal. Die Häufigkeit eines Wortes im akustischen Training ist ebenfalls mit den Fehlern korreliert.

Die zur Sprechgeschwindigkeit assoziierten Merkmale weisen eine verhältnismäßig geringe Korrelation zu den Fehlern auf, besitzen aber, wie die Graphen zeigen, einen nichtlinearen Zusammenhang zu den Fehlern. Es war möglich durch Übertragen von Wissen aus der akustische Trainingsmenge (mittlere Länge von Phonemen und Worten) die Korrelation zu den Fehlern zu verbessern.

Bei den Merkmalen der akustische Ähnlichkeit wurden verschiedenen Verfahren der Normalisierung untersucht. Dabei ergab sich, daß eine Normalisierung mit den a priori Wahrscheinlichkeiten eines vereinfachten Spracherkenners (kontextunabhängiger Phonemerkenner) am besten ist.

Als Merkmal für Unsicherheit in der Suche wurde die akustische Stabilität untersucht. Sie besitzt eine hervorragende Korrelation zu den Fehlern und mißt wie häufig ein zu bewertendes Wort in einer Matrix von Hypothesen mit geändertem Sprachmodellgewicht und Wortübergangsstrafterm auftritt. Als neues Maß für die Unsicherheit im Worthypothesengraph wurde die Entropie über Teilgraphen des Worthypothesengraphen herangezogen. Sie besitzt ebenfalls eine gute Korrelation zu den Fehlern. Verschiedene Berechnungsmodi wurden untersucht. Dabei zeigte sich, daß es für Einfügefehler, Ersetzungsfehler und OOV-Fehler drei verschiedene Modi gibt, die mit den jeweiligen Fehlern am besten korreliert sind, aber nicht völlig voneinander abhängen.

Um die Eigenschaften der Merkmale in Kombination zu untersuchen, wurde ein Neuronales Netz mit einer versteckten Einheit für die Aufgabe als Vertrauensmesser trainiert. Dazu wurden die Merkmale untersucht, die aufgrund der Korrelationen mit den Fehlern besonders aussichtsreich für die Konstruktion eines guten Vertrauensmessers erscheinen.

Zunächst wurden die Merkmale der Unsicherheit in der Suche (akustische Stabilität, Entropie im Worthypothesengraphen) verglichen, indem für jedes ein Neuronales Netz trainiert und anschließend das NIST-QM bestimmt wurde. Dabei zeigte sich, daß für sich genommen die akustische Stabilität dem Entropiemaß überlegen ist. Wird aber die oben genannte Kombination von drei Merkmalen aus der Entropie verwendet, sind diese verglichen mit der akustischen Stabilität fast gleich gut anzusehen. Daß sich bei der Kombination der Merkmale akustische Stabilität und Entropie im Worthypothesengraphen eine deutliche Verbesserung des NIST-QM ergeben hat, ist dabei besonders positiv zu bewerten.

Die weiteren Untersuchungen ergaben, daß die Merkmale der Sprechgeschwindigkeit ebenfalls zu einer deutlichen Verbesserung führten. Die Wissensquelle Sprechgeschwindigkeit wird vom untersuchten Spracherkenner noch nicht verwendet. In Zukunft kann dieses Merkmal an Qualität verlieren, wenn es einmal für die Berechnung der Hypothesen herangezogen wird. Die wortassoziierten Merkmale ergaben ebenfalls eine deutliche Verbesserung bezüglich des NIST-QM.

Daß die mittlere Wort-Score aus der akustischen Ähnlichkeit einen so negativen Effekt besitzt, ist erstaunlich und auf die Inhomogenität der Datenmengen zurückzuführen. Eine niedrige Score bei der mittleren Wort-Score wird häufig bei Buchstabierungsfehlern erreicht, die aber in der Evaluationsmenge nicht vorgekommen sind. Allgemein ist aber eine niedrige Score normalerweise (ohne Buchstabierung) ein Anzeichen für ein korrektes Wort. Da das NIST-QM die Fähigkeit die a posteriori Wahrscheinlichkeit $p(Korrekt|w)$ zu schätzen bewertet, wirkt sich dieser Umstand negativ aus. Das Ergebnis zeigt, daß bei der gegebenen Inhomogenität der Daten auf die akustische Ähnlichkeit verzichtet werden soll.

Weiter zeigten die Experimente, daß durch Hinzunahme des Kontextes des betrachteten Wortes (linkes + rechtes Nachbarwort) die Qualität eines Vertrauensmessers weiter gesteigert werden kann. Das beste Ergebnis beim NIST-QM war 0.291 und wurde mit einer Merkmalskombination aus den Bereichen wortassoziierte Merkmale, Sprechgeschwindigkeit, Unsicherheit im Suchraum und Hinzunahme des Kontext erreicht. Mit der gleichen Kombination wurden 80 % aller Worte (B-QM) bei einer Baseline von 68,3 % richtig markiert. Dies entspricht einer Fehlerreduktion von 36,9 %. Es konnte gezeigt werden, daß es möglich ist, Fehler eines Spracherkenners zu identifizieren und ein Vertrauensmaß für seine Hypothese anzugeben.

7.2 Ausblick

Nachdem eine Vertrauensmessung für den Janus-Spracherkenner möglich ist, stellen sich weitere Fragen.

Die wichtigste Frage ist, ob die als gut befundenen Merkmale auch auf neue Spracherkennungssysteme übertragen werden können. Bei einigen, beispielsweise der Häufigkeit im akustischen Training, ist zu erwarten, daß aufgrund verbesserter Modelle in der Spracherkennung die Aussagekraft über die Fehler geringer wird. Ob die akustische Stabilität und die Entropie im Worthypothesengraphen in verbesserten Spracherkennern genauso gut sind, ist zu überprüfen. Die Übertragbarkeit scheint aussichtsreich. Auch eine Frage der Übertragbarkeit ist, wie gut die Merkmale bei Spracherkennern mit kleinen Vokabularien (Kommandosätze, Ziffern) sind.

Ebenfalls wichtig ist die Frage der Anwendbarkeit eines Vertrauensmessers. Kann er wirklich sinnvoll für Parser, Dialogsysteme und zur Verbesserung der Spracherkennung eingesetzt werden? Hier stellt sich das Problem, daß es für fast alle Anwendungen sinnvoll ist, auch Löschfehler erkennen zu können. Dazu sind die Lücken einer Hypothese zu bewertet wozu vermutlich ein Zeitalign-Vertrauensmesser benötigt wird.

Es stellt sich die Frage, wie sich der Vertrauensmesser weiter verbessern läßt. Dies könnte durch noch ungenutzte Wissensquellen geschehen. Beispielsweise der *Signal zu Rauschabstand* bei geräuschvollen Umgebungen, wie in einem Auto. Eine andere Möglichkeit ist die Verbesserung der Klassifikatoren selbst. Beispielsweise könnten auch *Entscheidungsbäume* verwendet werden, die dann auch die Wortidentität berücksichtigen können.

Lassen sich die untersuchten Merkmale für eine Bewertung verbessern? Die normalisierte mittlere Wort-Score könnte durch verbesserte Modelle (kontextabhängige HMM) berechnet werden. Wird dies mit einem Spracherkenner durchgeführt kann sich ein großer Berechnungsaufwand ergeben. Durch spezielle Modelle kann möglicherweise eine schnelle Berechnung der a priori Wahrscheinlichkeit für die Normalisierung erreicht werden. Die Entropie im Worthypothesengraph ist wegen der vereinfachten Berechnungvorschrift eventuell noch verbesserungsfähig.

Da sich gezeigt hat, daß sich die Merkmale der Sprechgeschwindigkeit durch Übertragen von Wissen aus der akustischen Trainingsmenge verbessern lassen, wäre es nützlich zu wissen, ob sich noch mehr Wissen aus der akustischen Trainingsmenge übertragen läßt. Beispielsweise die mittlere "mittlere Wort-Score" eines Wortes.

Ein praktisches Problem der Spracherkennung ist es, genügend Daten zur Verfügung zu haben. Dies macht sich auch bei der Konstruktion eines Vertrauensmessers bemerkbar. Da sich eine Inhomogenität der Daten zur Zielanwendung sehr negativ auswirken kann, stellt sich die Frage nach einer Methode die Daten anzupassen.

Ist es möglich aus den erkannten Fehlern Einsicht über ihre Ursachen zu gewinnen und können diese behoben werden? Eine Fehleranalyse mit Hilfe eines Vertrauensmessers könnte hierüber Auskunft geben.

Abschließend stellt sich die Frage, ob die Entropie im Worthypothesengraphen auch auf Systeme übertragen werden kann, die nicht Spracherkennung betreiben, beispielsweise Schrifterkennung, und ob dies auch für nicht HMM-Systeme möglich ist, die aber eine ähnliche Struktur aufweisen (*MS-TDNN*, *Hybride Neuronale Netze* (HMM/NN)).

Literaturverzeichnis

[1] Günter Bamberg, Franz Baur: *Statistik*, Oldenbourg Verlag GmbH, München 1996

[2] Rainer Baumgärtner: *Kanalkompensation in der Spracherkennung*, Diplomarbeit am Institut Logik, Komplexität und Deduktionssystem, Universität Karlsruhe, Oktober 1996

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Riverst *Introduction to Algorithms*, The MIT Press, McGraw-Hill Book Company 1990

[4] Thomas M. Cover, Joy A. Thomas: *Elements of Information Theory*, Wilson & Sons, Inc, New York 1991

[5] S. Cox, R. Rose: *Confidence Measures for the Switchboard Database*, in Proc. ICASSP-96, pp. 511 ff, Atlanta, May 1996

[6] Steven B. Davis, Paul Mermelstein *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, 1980, in A. Waibel, K.-F. Lee: "Readings in Speech Recognition", Morgan Kaufmann Publishers, Inc., San Mateo 1990

[7] E. Eide, H. Gish, P. Jeanrenaud, A. Mielke: *Understanding and improving speech recognition performance through the use of diagnostic tools*, in Proc. ICASSP-95, pp. 221 ff, vol 1, Detroit, Michigan, May 1995

[8] M. Finke: persönliche Kommunikation

[9] Keinosuke Fukunaga: *Introduction to statistical pattern recognition 2nd ed.*, Academic Press, Inc., San Diego 1990

[10] Wolfgang Hürst: *Adaptive bimodale Sensorfusion für automatische Spracherkennung und Lippenlesen*, Studienarbeit am Institut Logik, Komplexität und Deduktionssystem, Universität Karlsruhe, Mai 1995

[11] F. Jelinek: *Self-Organized Language Modeling for Speech Recognition*, in A. Waibel, K.-F. Lee: "Readings in Speech Recognition", Morgan Kaufmann Publishers, Inc., San Mateo 1990

[12] T. Kemp, A. Jusek: *Modelling unknown words in spontaneous speech*, in Proc. ICASSP-96, pp. 530 ff, Atlanta, May 1996

[13] E. Lleida, R.C. Rose: *Efficient decoding and training procedures for utterance verification in continuous speech recognition*, in Proc. ICASSP-96, pp. 507 ff, Atlanta, Georgia, May 1996

[14] Udi Manber *Introduction to algorithm*, Addison Wesley Publishing Company 1989

[15] Hermann Ney, Xavier Aubert: *A Word Graph Algorithm For Large Vocabulary, Continuous Speech Recognition*, in Proc. ICSLP-94 pp. 1355 ff, Yokohama 1994

[16] Volker Steinbiss, Bach-Hiep Tran, Hermann Ney: *Improvements in Beam Search*, in Proc. ICSLP-94 pp. 2143 ff, Yokohama 1994

[17] Hermann Ney: *The Use of a One-Stage Dynamic Programming Algorithm for Connectet Word Recognition*, 1984, in A. Waibel, K.-F. Lee: "Readings in Speech Recognition", Morgan Kaufmann Publishers, Inc., San Mateo 1990

[18] T. Otsuki, A. Ito, S. Makino, T. Otomo: *The performance prediction method on sentence recognition system using a finite state automaton*, in Proc. ICASSP-94, pp. I-397 ff, Adelaide, Australia, April 1994

[19] T. Ottmann, P. Widmayer *Algorithmen und Datenstrukturen* BI-Wissenschaftsverlag, Mannheim 1990

[20] John K. Ousterhout *Tcl and the Tk Toolkit*, Addison Wesley Publishing Company, Massachusetts 1995

[21] Haitao Qiu: *Confidence Measure for Speech Recognition Systems*, Masters Thesis, Carnegie Mellon University Computational Linguistics Philosophy Department, Pittsburgh, PA, April 1996

[22] Lawence R. Rabiner *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, 1988, in A. Waibel, K.-F. Lee: "Readings in Speech Recognition", Morgan Kaufmann Publishers, Inc., San Mateo 1990

[23] Lawence R. Rabiner, Stephen E. Levison *Isolated and Connected Word Recognition–Theory and Selected Applications*, 1981, in A. Waibel, K.-F. Lee: Morgan Kaufmann Publishers, Inc., San Mateo 1990

[24] L. R. Rabiner, R. W. Schafer: *Digital Processing of Speech Signals*, Prentice-Hall, Inc., New Jersey 1978

[25] L. R. Rabiner, R. W. Schafer: *Digital Representations of Speech Signals*, 1984, in A. Waibel, K.-F. Lee: "Readings in Speech Recognition", Morgan Kaufmann Publishers, Inc., San Mateo 1990

[26] M. Rahim, C.H. Lee, B.H. Juang: *Robust utterance verification for connected digits recognition*, in Proc. ICASSP-95, pp. 285 ff, Detroit, Michigan, May 1995

[27] Z. Rivlin, M. Cohen, V. Abrash, T. Chung: *A phone-dependent confidence measure for utterance rejection*, in Proc. ICASSP-96, pp. 515 ff, Atlanta, Georgia, May 1996

[28] Raúl Rojas: *Theorie der Neuronalen Netze*, Springer-Verlag. Berlin, Heidelberg 1993

[29] R.C. Rose, B.H. Juang, C.H. Lee: *A training procedure for verifying string hypotheses in continuous speech recognition*, in Proc. ICASSP-95, pp. 281 ff, Detroit, Michigan, May 1995

[30] Claude E. Shannon, Warren Weaver: *Mathematische Grundlagen der Informationstheorie*, R. Oldenbourg Verlag GmbH, München 1976

[31] S.R. Young, W. Ward: *Recognition confidence measures for spontaneous spoken dialog*, in Proc. EUROSPEECH 93, pp. 1177 ff, Berlin, Germany, September 1993

[32] Sheryl Young: *Detecting misrecognitions and out-of-vocabulary words*, in Proc. ICASSP-94, pp. II-21 ff, Adelaide, Australia, April 1994

[33] Ernst Günter Schukat-Talamazzini: *Automatische Spracherkennung*, Vieweg, Braunschweig/Wiesbaden 1995

[34] R.A. Sukkar: *Rejection for connected digit recognition based on GPD segmental discrimination*, in Proc. ICASSP-94, pp. I-393 ff, Adelaide, Australia, April 1994

[35] R.A. Sukkar: *Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training*, in Proc. ICASSP-96, pp 518 ff, Atlanta, Georgia, May 1996

[36] Johannes Volmert (Hrsg.) *Grundkurs Sprachwissenschaft* Willhelm Fink Verlag München 1995

[37] Alex Waibel, Michael Finke, Thomas Kemp, Donna Gates, Marsal Gavaldà, Arthur McNair, Alon Lavie, Lori Levin, Laura Mayfield, Martin Maier, Ivica Rogina, Kaori Shima, Tilo Sloboda, Monika Woszczyna, Puming Zhan, Torsten Zeppenfeld *JANUS-II — Translation of Spontaneous Conversational Speech* in Proc. ICSLP-96 pp. 409 ff, Atlanta, Georgia 1996

[38] Alex Waibel, Kai-Fu Lee: *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo 1990

Anhang A

Berechnung der LatEntropie

Dieser Anhang beschreibt die Berechnung der Entropie in der Suche, das Merkmal nennen wir LatEntropie. Die vorausgesetzte Struktur einer Lattice des Janus-3 ist in Abschnitt 2.10 beschrieben. Grundlegende Überlegungen zur Berechnung der Entropie in einem Graphen finden sich in Abschnitt 2.9.

Es sei eine Lattice L gegeben, die über eine gesamte Äußerung erzeugt wurde, die Hypothese $W = w_1 \ldots w_n$ mit den dazugehörigen Zeitsegmenten $[t_a^{w_1}, t_e^{w_1}]; \ldots; [t_a^{w_n}, t_e^{w_n}]$ sei aus dieser Lattice gewonnen. Da für jedes einzelne Wort der Hypothese die Glaubwürdigkeit bestimmt werden soll, ist es sinnvoll, auch den diesem Wort entsprechenden Zeitbereich zu untersuchen. Da die Lattice aber für eine gesamte Äußerung bestimmt wird, ergibt sich nun die Frage, wie ein Teilgraph für die Berechnung der LatEntropie eines einzelnen Wortes bestimmt werden kann.

Soll beispielsweise für das Wort w_i der Hypothese die Berechnung des Merkmals durchgeführt werden, kann zunächst mit Hilfe der Zeitzuordnung $[t_a^{w_i}, t_e^{w_i}]$ die Lattice in die drei Bereiche vor w_i (A), während w_i (B) und nach w_i (C) unterteilt werden. Abbildung A.1 zeigt eine Lattice mit eingezeichneten Zeitgrenzen.

Aus der Abbildung ist ersichtlich, daß Kanten die Zeitgrenzen t_a und t_e schneiden können. Mit der Bereichseinteilung lassen sich sechs Typen von Kanten unterscheiden, die in Tabelle A.1 angegeben sind.

Einige Kanten in Abbildung A.1 sind mit ihren Kantentypen beschriftet. Für die weiteren Betrachtungen sind dabei die Kanten am wichtigsten, die in irgendeiner Weise den betrachteten Zeitraum $[t_a, t_e]$ berühren (Typ b, d, e, f).

Anhand der gegebenen Zeitgrenzen können die Knoten in fünf Typen unterteilt werden, die in Tabelle A.2 angegeben sind. Von besonderer Bedeutung dabei sind die Knoten, die genau auf einer Zeitgrenze liegen (Typ 0, 1).

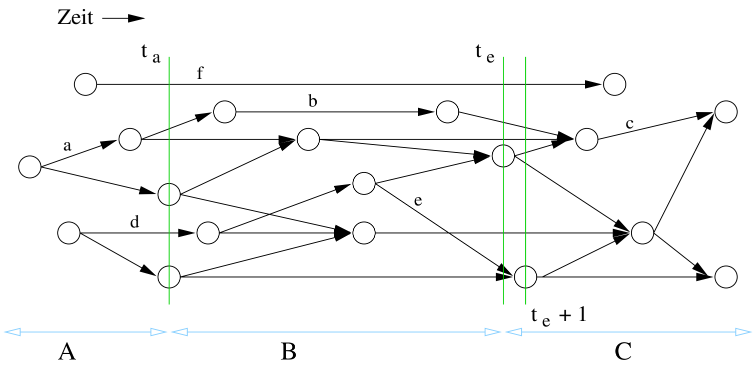


Abbildung A.1: Lattice L mit Zeitbereichen und Kantentypen

Typ	Kante liegt
a	ganz im Bereich A
b	ganz im Bereich B
c	ganz im Bereich C
d	in den Bereichen A und B
e	in den Bereichen B und C
f	in den Bereichen A, B und C

Tabelle A.1: Kantentypen

Typ	Knoten liegt
0	auf Zeitpunkt t_a
1	auf Zeitpunkt t_e
2	vor Zeitpunkt t_a
3	zwischen Zeitpunkt t_a und t_e
4	nach Zeitpunkt t_e

Tabelle A.2: Knotentypen

Anhand des folgenden Pseudocode-Programms wird die grundlegende Berechnungsvorschrift der LatEntropie dargestellt.

```
Funktion: CalcLatEntropy
Eingabe:  Lattice L = (V,E) // Knoten V und Kanten E
          Startzeit t_a
          Endzeit t_e
Ausgabe:  Entropie des Teilgraphen, dessen Knoten
          zwischen t_a und t_e liegen

// 1. Konstruktion des Teilgraphen
SVM := Menge der Knoten v[i] mit v[i].start = t_a
TEM := Menge der Kanten e[i,j] mit t_a <= v[i].start <= t_e
TVM := Menge der Knoten, die Anfang oder Ende einer Kante aus TEM sind

v[neu] := neuer Knoten
fuer alle Knoten v[i] aus SVM
  SEM := SEM vereinigt mit Kante e[neu,i] = (v[neu],v[i])

E'  := Kantenmenge TEM vereinigt mit Kantenmenge SEM
V'  := Knotenmenge TVM vereinigt mit { s[neu] }
L'  := Graph (V',E')

// 2. Berechnung der Entropie
Entropie := H(L',v[neu])
Return Entropie
```

Im Pseudocode-Programm wird die Berechnung der Entropie gemäß der Darstellung in Abschnitt 2.10 durchgeführt, die kurzgefaßt folgender rekursiver Formel entspricht:

$$
\begin{aligned}
L &= (V, E) \text{ Lattice aus Knoten V und Kanten E} \\
(L, v_i) &= \text{Subgraph in Lattice L mit } v_i \text{ als Startknoten} \\
H(L, v_i) &= \underbrace{\sum_{(v_i,v_j)\in E} (v_i, v_j).prob * \log_2 (v_i, v_j).prob}_{\text{Entropie des betrachteten Knoten}} + \underbrace{\sum_{(v_i,v_j)\in E} (v_i, v_j).prob * H(L, v_j)}_{\text{Entropie der Subgraphen}}
\end{aligned}
$$

Die vorgestellte Art der Berechnung wirft einige Schwierigkeiten auf. Zunächst soll die Konstruktion des Teilgraphen näher betrachtet werden. Um die Konstruktion zu verdeutlichen, ist in Abbildung A.2 der resultierende Teilgraph L' des Graphen L aus Abbildung A.1 dargestellt. Die gestrichelten Kanten und Knoten sind bei der Konstruktion weggefallen. Für Kanten des Typs a und c ist dies auch gewünscht, da sie außerhalb des betrachteten Zeitbereichs liegen. Daß dagegen

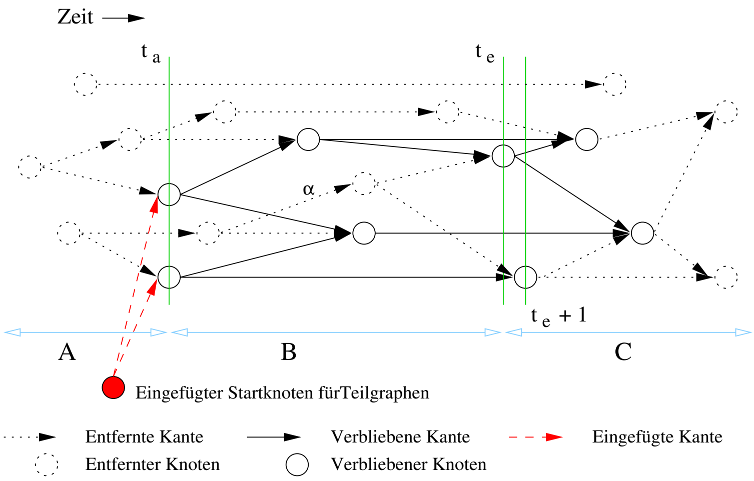


Abbildung A.2: Teilgraph L' der Lattice L

beispielsweise die mit α gekennzeichnete Kante wegfällt, erscheint nicht einsichtig, da es anscheinend einen Pfad gibt, der durch den betrachteten Zeitbereich führt und unberücksichtigt bleibt. Ein solcher Pfad bedeutet aber eine zusätzliche Unsicherheit für diesen Zeitbereich.

Verschiedene Ansätze zur Lösung dieser Schwierigkeit sind denkbar:

1. Beschränken der Kanten auf den betrachteten Zeitbereich. Wenn eine Kante e über eine Zeitgrenze hinausgeht, wird der im Bereich A oder C liegende Knoten auf die nächste Zeitgrenze verschoben. Es ist dann auch die Wahrscheinlichkeit $e.prob$ anzupassen.

2. Einführen eines Parameters $delta$, der aus der Zeitgrenze t_a den Zeitbereich $S = [t_a - delta, t_a + delta]$ macht. Alle Knoten, die in diesen Zeitbereich fallen, werden zur Menge SVM des Pseudocode-Programms, zusätzlich wird t_e ebenfalls um den Wert $delta$ vergrößert. Knoten, die nun nahe dem Zeitpunkt t_a liegen, und deren Subgraphen werden ebenfalls in die Berechnung einbezogen.

Das $delta$ des zweiten Ansatzes läßt sich anschaulich auch so verstehen, daß über die exakten Zeitgrenzen Unsicherheit besteht. Nachteil dieses Ansatzes ist zum einen, daß nicht sichergestellt ist auch alle möglichen Pfade zu erfassen, zum anderen, daß ein zusätzlicher Parameter benötigt wird, dessen Größe empirisch zu bestimmen ist. Dieser Ansatz wurde realisiert, da beim ersten Ansatz unklar ist, wie die neuen Übergangswahrscheinlichkeiten geschätzt werden sollen.

Eine im Pseudocode-Programm offen geblieben Frage ist, wie für eine Kante e die Wahrscheinlichkeit $e.prob$ bestimmt wird. Die Struktur einer Lattice in

Janus-3 bietet mit der lscore und der pscore einer Kante e mehrere Möglichkeiten an.

Da Scores negativ logarithmierte Wahrscheinlichkeiten sind und es nicht sichergestellt ist, daß sich die Wahrscheinlichkeiten der Kanten eines Knoten zu 1.0 addieren, müssen Scores mit folgender Formel in normalisierte Wahrscheinlichkeiten umgerechnet werden

$$e_{i,j}.prob = \frac{exp(-e_{i,j}.score)}{\sum_{e_{i,k} \in E} exp(-e_{i,k}.score)} \tag{A.1}$$

Für die Berechnung der Kantenscore $e.score$ sind in Tabelle A.3 verschiedene Berechnungsmodi angegeben. Am wichtigsten sind dabei die Modi 0, 2 und 4, weil sie die Wahrscheinlichkeit annähern, daß an der entsprechenden Kante die Emission des zugehörigen Wortes stattgefunden hat. Dabei nähert Modus 0 dies aufgrund der akustischen Score an, Modus 4 soll dagegen nur den Sprachmodellanteil berücksichtigen und Modus 2 kombiniert beide Scores. Die restlichen Modi sind experimentell und der Vollständigkeit halber angegeben. Eine Normierung der Scores mit der Wortdauer führt dazu, daß die Wahrscheinlichkeiten von der Wortlänge unabhängig werden. Das kann zur Folge haben, daß sich die Scores kurzer und langer Worte angleichen. Insgesamt wächst dadurch die Entropie in einem Knoten und die Unsicherheiten der Subgraphen kurzer Kanten haben in Formel A.1 ein geringeres Gewicht gegenüber den Subgraphen langer Kanten.

Modus	Scoreberechnung
0	$e_{i,j}.score = e_{i,j}.lscore$
1	$e_{i,j}.score = e_{i,j}.lscore/(v_j.start - v_i.start)$
2	$e_{i,j}.score = e_{i,j}.pscore$
3	$e_{i,j}.score = e_{i,j}.pscore/(v_j.start - v_i.start)$
4	$e_{i,j}.score = (e_{i,j}.pscore - \min e_{k,i}.pscore - e_{i,j}.lscore)$
5	$e_{i,j}.score = (e_{i,j}.pscore - \min e_{k,i}.pscore)/(v_j.start - v_i.start)$
6	$e_{i,j}.score = (e_{i,j}.pscore - \min e_{i,k}.pscore)$
7	$e_{i,j}.score = (e_{i,j}.pscore - \min e_{i,k}.pscore)/(v_j.start - v_i.start)$

Tabelle A.3: Berechnungsmodi für die Kantenscore

Eine Decodierung mit dem Viterbi-Algorithmus kann sowohl in als auch gegen die Zeitrichtung durchgeführt werden. In beiden Fällen ergibt der beste Viterbi-Pfad die gleichen Scores und besitzt die gleiche Zustandsfolge[1]. Es ist nicht unbedingt notwendig die Berechnungsrichtung der LatEntropie immer in Zeitrichtung

[1] Der Suchraum darf für diese Aussage nicht beschnitten werden.

zu wählen. Um die Richtung zu bestimmen, gibt es ein Richtungsflag, das festlegt, ob der Teilgraph der Lattice in oder gegen die Zeitrichtung erzeugt wird. Wird der Teilgraph gegen die Zeitrichtung berechnet, nimmt t_e im Pseudocode die Rolle von t_a ein und die Kanten sind gegen die Zeitrichtung orientiert. Dabei ist zu bemerken, daß die pscores in Zeitrichtung addiert wurden und dementsprechend auch das Sprachmodell diese Orientierung besitzt. Die Berechnung der Kantenscore $e_{i,j}$ ist deshalb in den Fällen zu negieren, in denen die Differenz mit einer Vorgängerkante sonst zu einer negativen Score führt.

Da es sich bei den Scores aus der Lattice, wie bereits erwähnt, um Wahrscheinlichkeiten von Viterbi-Pfaden handelt, geben sie nicht an, wie wahrscheinlich es ist, daß irgendein Pfad über eine bestimmte Kante führt. Erst wenn die Wahrscheinlichkeit der Kante angibt, daß in den nachfolgenden Subgraphen verzweigt wird, ist eine korrekte Entropieberechnung möglich.

Mit Hilfe der Wahrscheinlichkeit $(p(i,j))$, daß irgendein Pfad über eine bestimmte Kante $e_{i,j}$ führt, kann berechnet werden, daß von einem *bestimmten* Knoten aus, beispielsweise v_i, über eine Kante, beispielsweise $e_{i,j}$ in den nachfolgenden Subgraphen verzweigt wird. Folgende Formel berechnet $p(i,j)$

$$p(i,j) = \frac{\alpha * \gamma * \beta}{\Pi} \tag{A.2}$$

$$\begin{aligned} & && \text{Wahrscheinlichkeit, daß} \\ \alpha &= && \text{irgendein Pfad vom Startknoten bis zum Knoten } v_i \text{ führt} \\ \beta &= && \text{irgendein Pfad vom Endknoten bis zum Knoten } v_j \text{ führt} \\ \gamma &= && \text{eine bestimmte Observation gemacht wurde} \\ \Pi &= && \text{irgendein Pfad, der im Startknoten beginnt, im Endknoten endet} \end{aligned}$$

Diese Formel wird auch im Baum-Welch-Algorithmus verwendet [22]. Eine Implementierung für die Berechnung der Übergangswahrscheinlichkeiten, die sich aus diesen Überlegungen ergeben, wurde aus Zeitgründen nicht durchgeführt. Das rechtfertigt sich nur dadurch, daß bereits mit den vereinfachten Berechnungsvorschriften eine Diskriminierung zwischen korrekt und nicht korrekt erkannten Worten möglich war.

Um zu verdeutlichen, welchen Effekt die vereinfachte Berechnungsvorschrift hat, soll ein Beispiel betrachtet werden. Abbildung A.3 zeigt eine einfache Lattice, deren Kanten mit Wahrscheinlichkeiten beschriftet sind. Dabei sollen die Wahrscheinlichkeiten aussagen wie gut das Modell der Suche gepaßt hat. Betrachten wir die Lattice als Sender für mögliche Hypothesen, dann ist die Entropie ein Maß für die mittlere Wahlfreiheit des Senders. Ist der Sender eine ergodische Quelle,

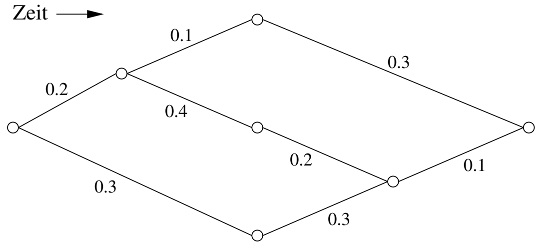


Abbildung A.3: Lattice mit lokalen Wahrscheinlichkeiten

so kann beliebig gut auf die Struktur der Quelle geschlossen werden, wenn wir im Besitz einer hinreichend langen Nachricht der Quelle sind. Angenommen wir besitzen eine solche Nachricht W und betrachten nun die dazu reverse[2] Nachricht W^r, dann ist aufgrund des einfachen und vor allem *deterministischen* Zusammenhangs der beiden Nachrichten W und W^r davon auszugehen, daß die Quellen, die beide Nachrichten produzieren, die gleiche mittlere Wahlfreiheit (Entropie) besitzen. Eine Nachricht W^r kann auch dadurch gewonnen werden, daß die ursprüngliche Quelle umgeformt wird. Hierzu sind in der Lattice die Kanten gegen die Zeitrichtung zu orientieren. Abbildung A.4 links und rechts zeigen jeweils eine Lattice bei der die Wahrscheinlichkeiten der Kanten normalisiert wurden. In der linken Abbildung wurden die Kanten zuvor in Zeitrichtung, in der rechten gegen die Zeitrichtung orientiert.

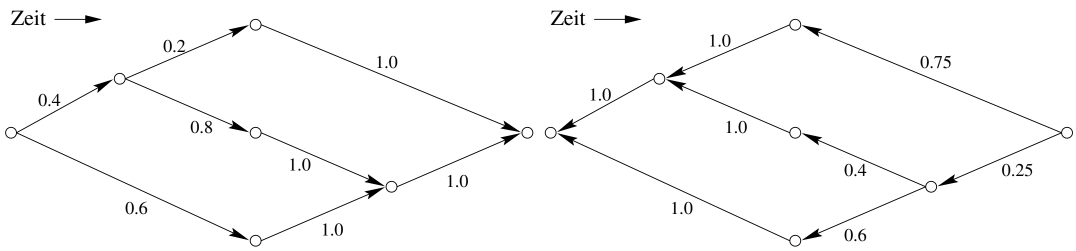


Abbildung A.4: Normalisierte Lattice in/gegen Zeitrichtung (links/rechts)

Nach obiger Ausführung, sollte die Entropie beider Graphen gleich sein, das ist aber nicht der Fall. Die Entropie des Graphen der linken Abbildung beträgt 1,693 Bit, die des Graphen der rechten Abbildung dagegen 1.782 Bit. Der Grund ist, daß es sich bei den Wahrscheinlichkeiten der Kanten nur um *lokale* Wahrscheinlichkeiten aus der Suche handelt, die die Subgraphen nicht berücksichtigen.

Dagegen bezieht der Baum-Welch-Algorithmus die Subgraphen mit ein. Abbildung A.5 links zeigt den Graphen aus Abbildung A.3, bei dem an den Kanten die Werte für α und β eingetragen sind. Rechts wurden die Kanten mit den aus Formel A.2 resultierenden Übergangswahrscheinlichkeiten beschriftet. Wird nun

[2] einfach rückwärts gelesen

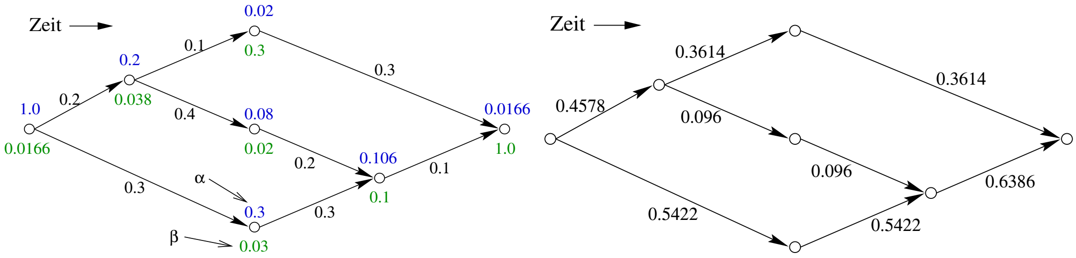


Abbildung A.5: Baum-Welch: Zwischenergebnis (links) und Ergebnis (rechts)

dieser Graph wie bereits die anderen orientiert und die Wahrscheinlichkeiten der Kanten normalisiert, ergeben sich die Graphen in Abbildung A.6 links und rechts. *Beide* Graphen besitzen eine Entropie von 1.334 Bit. Der Grund ist, daß durch den Baum-Welch-Algorithmus und die anschließende Normalisierung der Wert der Kante nun angibt, wie wahrscheinlich es ist, über sie in den Subgraphen zu verzweigen.

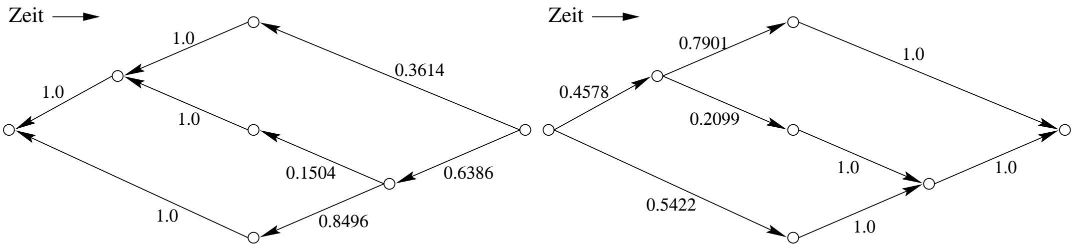


Abbildung A.6: Normalisierte Lattice aus Baum-Welch in/gegen Zeitrichtung (links/rechts)

Anhang B

Zusammenfassung: Korrelationen

Tabelle B.1 faßt die gefundenen Korrelationen aus Kapitel 5 zusammen. Die Reihenfolge ist nach absteigender Korrelation zum Gesamtfehler geordnet.

	Gesamt-FR	Einfüge-FR	OOV-FR
AStabil	-54,0 %	-20,6 %	-18,1 %
H1	47,9 %	18,4 %	16,0 %
H2	46,9 %	18,6 %	16,5 %
H3	35,5 %	11,9 %	19,5 %
SM-NGRAM	-19,6 %	-0,7 %	-14,3 %
AMWScore	13,5 %	8,8 %	5,9 %
PMWScore	13,1 %	9,5 %	3,9 %
MWScore	12,4 %	7,6 %	2,6 %
LogPhonAnz	-11,8 %	-12,1 %	-1,0 %
EMWScore	10,7 %	8,5 %	11,8 %
LogAnzATrain	-10,7 %	-0,3 %	-16,2 %
PhonAnz	-10,3 %	-9,8 %	-0,5 %
WStreck3	6,6 %	6,0 %	11,5 %
WStreck2	5,0 %	6,0 %	10,1 %
WStreck1	3,5 %	5,8 %	9,4 %
WStauch1	-2,4 %	-3,6 %	-7,9 %
WStauch2	-2,4 %	-2,7 %	-8,7 %
WSchw	2,3 %	1,3 %	1,0 %
WStauch3	-1,3 %	-1,6 %	-8,2 %
KSchw	-0,5 %	-0,7 %	1,2 %
AnzATrain	-0,4 %	-0,3 %	-0,8 %

Tabelle B.1: Zusammenfassung der Korrelationen