„Statistische Auswahl von Wortabhängigkeiten in der automatischen Spracherkennung“

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Informatiker Sven Carl Martin

aus

Köln

Berichter: **Universitätsprofessor Dr.–Ing. Hermann Ney**
Universitätsprofessor Dr. Wolf Paprotté

Tag der mündlichen Prüfung: 03. Februar 2000

Knowledge is gained by learning;
trust by doubt;
skill by practice;
and love by love.

— T. S.

Danksagung

Diese Arbeit entstand während meiner Tätigkeit am Lehrstuhl für Informatik VI der Rheinisch Westfälischen Technischen Hochschule Aachen. Dem Lehrstuhlinhaber, Herrn Univ.–Prof. Dr.–Ing. Hermann Ney, möchte ich an dieser Stelle für seine sehr intensive Betreuung, vielfältigen Anregungen und kritischen Betrachtungen herzlich danken. Mein Dank gilt auch Herrn Univ.–Prof. Dr. Wolf Paprotté vom Arbeitsbereich Linguistik der Westfälischen Wilhelms–Universität Münster für die Übernahme des Koreferats. Sehr dankbar bin ich auch den Kollegen und Studenten am Lehrstuhl für Informatik VI für die gute Zusammenarbeit und die vielen Diskussionen in der Vergangenheit. Besonders möchte ich meinen Eltern und meinem Freundeskreis für die moralische Unterstützung der letzten Jahre danken, die den Abschluß dieser Arbeit erst möglich gemacht hat.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einführung

Die statistische Verarbeitung natürlicher Sprache hat in den letzten Jahren zunehmend an Bedeutung gewonnen. Ausgehend von den ersten Arbeiten in den Bereichen Spracherkennung [Bahl et al. 83] und automatische Übersetzung [Brown et al. 90] sind die statistischen Ansätze mittlerweile für große Anwendungen mit einem Wortschatz von mehreren tausend Wörtern in der Spracherkennung [Klakow 98a, Woodland et al. 98] und der automatischen Übersetzung [Berger et al. 94, Tillmann et al. 97] erfolgreich eingesetzt worden und haben, zumindest in der Spracherkennung, in jüngster Zeit ihren Weg in die kommerzielle Anwendung gefunden. Dies wurde möglich durch die in den letzten Jahren deutlich gestiegenen Rechen– und Speicherkapazitäten einerseits und die konzeptionelle Weiterentwicklung und Optimierung der beteiligten Algorithmen andererseits.

Ein wesentlicher Bestandteil sowohl der Spracherkennung als auch der automatischen Übersetzung ist das sogenannte linguistische oder Sprachmodell zur Bestimmung der a–priori–Wahrscheinlichkeit von Wortfolgen unabhängig von Akustik und Übersetzung [Jelinek 91]. Diese Wahrscheinlichkeit wird nach dem Maximum–Likelihood–Kriterium aus einer Statistik über ausgewählte Eigenschaften eines Trainingstextes gewonnen, wobei Maximum–Likelihood hier die Optimierung der Wahrscheinlichkeiten auf diesem Trainingstext bedeutet. Dabei sind zwei Probleme zu lösen:

a. Die Auswahl dieser Eigenschaften: Zu Beginn dieser Arbeit hatte sich gerade das sogenannte Trigramm–Sprachmodell etabliert. Es handelt sich dabei um eine Statistik über die im Trainingstext gesehenen aufeinanderfolgenden Worttripel, eine konzeptionell sehr einfache, aber in der Anwendung auch sehr effektive Modellierung. Es war offen, um welche weitere Eigenschaften das Trigramm–Sprachmodell sinnvoll ergänzt werden konnte.

b. Die Behandlung von im Training unbeobachteten Ereignissen in der Anwendung: Über solche Ereignisse, im Fall des Trigramm–Sprachmodells also Worttripel, die im Traingstext nicht vorkommen, kann natürlich im Training keine Statistik geführt werden. Nach dem Maximum–Likelihood–Kriterium haben sie daher eine geschätzte Wahrscheinlichkeit Null, obwohl es sich durchaus um sinnvolle Ereignisse handeln kann. Zu Beginn dieser Arbeit war es üblich, in einem solchen Fall mit

Hilfe von Varianten der linearen Interpolation auf Statistiken über gröbere Ereignisse zurückzugehen, im Falle des Trigramm–Sprachmodells also Wortpaare, noch gröber Einzelwörter. Man spricht in so einem Fall von „Glättung“. In der Literatur gab es bereits Vorschläge für fortgeschrittene Glättungsverfahren, die jedoch nur an kleineren Aufgaben getestet worden waren.

Damit ergeben sich auch die beiden hauptsächlichen Aufgabenstellungen dieser Arbeit:

1. Es sollen die vorgeschlagenen Glättungsverfahren auf Aufgaben mit großen Trainingstexten und großem Vokabular untersucht werden. Da die Glättungsproblematik in realistischen Aufgaben, auch bei um weitere Eigenschaften ergänzten Trigramm–Sprachmodellen, immer gegeben ist, soll sie als wesentliches Problem zuerst behandelt werden.

2. Als Ergänzungen bieten sich an und sind in der Literatur teilweise schon beschrieben worden:

   - **Varigramme:** Im Gegensatz zum Trigramm–Sprachmodell werden selektiv auch Wortfolgen einer Länge größer drei betrachtet.
   - **Wortphrasen:** Geeignete Wortpaare oder auch längere Wortfolgen werden zu neuen Wörtern (Wortphrasen) zusammengesetzt und das Vokabular des Trigramm–Sprachmodells entsprechend erweitert.
   - **Wortklassen:** Alle Wörter des Vokabulars werden geeigneten Wortklassen zugeordnet und die Statistik nicht mehr über die Worttripel, sondern über diese Wortklassentripel geführt.
   - **Abstand–Trigramme:** Die Statistik wird nicht mehr über aufeinanderfolgende Worttripel geführt, sondern über Worttripel mit einer Lücke nach dem ersten oder zweiten Wort.

   Dabei sollen folgende Randbedingungen gelten:

   - Diese Ergänzungen sollen ebenfalls auf Aufgaben mit großen Texten und Vokabularien untersucht werden.
   - Mit Ausnahme der Abstand–Trigramme ist zu beachten, daß der Anwendung erst eine Auswahl der geeigneten Varigramme, Wortphrasen und Wortklassen voraus gehen muß. Diese Auswahl soll wie die Schätzung der Wahrscheinlichkeiten nach dem Maximum–Likelihood–Kriterium erfolgen.
   - Die Auswirkung der gemeinsamen Anwendung dieser Erweiterungen ist nur unzureichend in der Literatur beschrieben und muß daher auch untersucht werden.

Die Untersuchungen werden sich auf die Spracherkennung beschränken, obwohl eine Verwendung der vorgeschlagenen Methoden für Sprachmodelle in der statistischen Übersetzung genauso möglich ist.

Diese Arbeit gliedert sich wie folgt: In **Kapitel 2** werden die mathematischen Grundlagen der Spracherkennung erläutert. Auch wird detailliert auf den Stand der Wissenschaft zu den oben genannten Punkten eingegangen und die Aufgabenstellung weiter spezifiziert. In den **Kapiteln 3 bis 7** werden jeweils für die Glättung und die vier vorgeschlagenen Erweiterungen die mathematischen Grundlagen, soweit nötig ihre Implementierungsaspekte und schließlich die erzielten Ergebnisse vorgestellt. **Kapitel 8** beinhaltet die Zusammenfassung der Ergebnisse und ihre Interpretation bezüglich der Aufgabenstellung aus Kapitel 2. **Anhang A** beschreibt die in dieser Arbeit verwendeten Korpora und **Anhang B** die verwendete Notation. Die weiteren Anhänge beschreiben im Detail mathematische Herleitungen und ergänzende Untersuchungen, die im Rahmen dieser Arbeit durchgeführt wurden.

Kapitel 2

Sprachmodellierung

2.1 Grundlagen

In Kapitel 2.1 wird ein Überblick über die Grundlagen der statistischen Spracherkennung und Sprachmodellierung gegeben. Das Training der Sprachmodelle nach dem Maximum–Likelihood–Kriterium wird vorgestellt, und mit Perplexität und Wortfehlerrate werden zwei Kriterien zur Bewertung von Sprachmodellen angegeben.

2.1.1 Statistische Spracherkennung

Das von der Schule vermittelte Idealbild der Sprache ist das eines durch Grammatik– und Ausspracheregeln genau festgelegten Gebildes. In der Realität ist die gesprochene Sprache jedoch durch die individuelle Realisierung der Laute, das Sprechtempo und den Sprachstil eines Sprechers geprägt. Eine zur automatischen Erkennung geeignete regelbasierte deterministische Grammatik oder ein anderes explizites Regelwerk ohne Wahrscheinlichkeiten zur Beschreibung der natürlichen Sprache, geschweige ihrer individuellen Ausprägungen, ist bisher nicht gefunden worden. Die Vielzahl der nur ungenau einzugrenzenden (z.B. durch den fließenden Übergang zwischen zwei ähnlichen Lauten) Bestandteile der Sprache und ihrer gegenseitigen Abhängigkeiten lassen dies auch als unwahrscheinlich erscheinen.

Statistische Beschreibungen (Modelle) der Sprache verzichten auf eindeutige Zuordnungen und lassen alle möglichen Interpretationen zu, nur eben mit unterschiedlicher Präferenz. Sie können daher viel flexibler mit den Ungenauigkeiten der gesprochenen Sprache umgehen als deterministische Regelwerke. Darüberhinaus steht ihnen der wissenschaftlich fundierte und bewährte Apparat der Stochastik zur Verfügung. In ihm gibt es mit den Wahrscheinlichkeiten eine normierte Bewertung der verschiedenen Abhängigkeiten, genaue Regeln zur Bestimmung dieser Wahrscheinlichkeiten durch statistisches Lernen sowie eine statistische Entscheidungstheorie, falls doch einmal eine harte Auswahl unter verschiedenen Möglichkeiten getroffen werden soll. Dabei gibt die Stochastik mit ihrem

Apparat nur den Rahmen vor, die eigentliche Modellierung muß nach wie vor problembezogen erfolgen.

Die Verwendung statistischer Modelle zur Spracherkennung geht jedoch über die üblicherweise gelehrte Statistik hinaus und eröffnet wissenschaftlich bisher wenig untersuchte Gebiete:

- Das statistische Lernen umfasst nicht nur die Bestimmung der Wahrscheinlichkeiten, sondern auch die Auswahl der zugrundeliegenden statistischen Modelle.
- Macht die Statistik üblicherweise nur Aussagen über vergangene Ereignisse, so soll sie nun zur Vorhersage ungesehener Daten, also des weiteren Verlaufs des Gesprochenen, herangezogen werden.
- Für diese Vorhersage ist auch eine brauchbare Modellierung seltener Ereignisse notwendig, die üblicherweise vernachlässigt werden.

In der konkreten Modellierung soll der statistische Spracherkenner einer beobachteten Signalfolge $x_1 ... x_T$ (kurz: x_1^T) der Länge T eine Wortfolge $w_1 ... w_N$ (kurz: w_1^N) der Länge N möglichst fehlerfrei zuordnen. Dazu wird die aus der statistischen Entscheidungstheorie bekannte Bayessche Entscheidungsregel verwendet [Bahl et al. 83], die lautet: Für eine beobachtete Folge von Messungen des akustischen Signals x_1^T ergibt sich im statistischen Mittel die kleinste Fehlerrate, falls die unbekannte Wortfolge w_1^N so bestimmt wird, daß

$$\begin{aligned} w_1^N &= \underset{w'^N_1}{\operatorname{argmax}} \left\{ Pr(w'^N_1 | x_1^T) \right\} \\ &= \underset{w'^N_1}{\operatorname{argmax}} \left\{ Pr(w'^N_1) \cdot Pr(x_1^T | w'^N_1) \right\} \quad . \end{aligned}$$

In die Entscheidungsregel gehen somit zwei Wahrscheinlichkeitsverteilungen ein:

- Die linguistische Wahrscheinlichkeit $Pr(w_1^N)$ als a-priori-Verteilung für alle möglichen Wortfolgen w_1^N.
- Die akustische Wahrscheinlichkeit $Pr(x_1^T | w_1^N)$ als a-posteriori-Verteilung für alle möglichen Signalfolgen x_1^T gegeben eine Wortfolge w_1^N.

Die tatsächlichen Verteilungen sind unbekannt und werden durch ein Sprachmodell bzw. ein akustisches Modell angenähert. Die Wortfolge w_1^N wird mit Hilfe eines Suchprozesses aufgrund der Bayesschen Entscheidungsregel ausgewählt. Zusammen mit der akustischen Analyse als Vorverarbeitung sind dies die Hauptbestandteile des statistischen Spracherkenners, der in Abb. 2.1 dargestellt ist. In der Praxis wird die linguistische Wahrscheinlichkeit $Pr(w_1^N)$ noch mit einem Skalierungsfaktor („LMScale“) potenziert.

2.1.2 Maximum–Likelihood–Parameterschätzung von Trigramm–Sprachmodellen

Die unbekannte linguistische a-priori-Verteilung $Pr(w_1^N)$ wird durch ein statistisches Sprachmodell $p_\theta(w_1^N)$ mit Parametersatz θ angenähert. Die Bayessche Entscheidungsre-

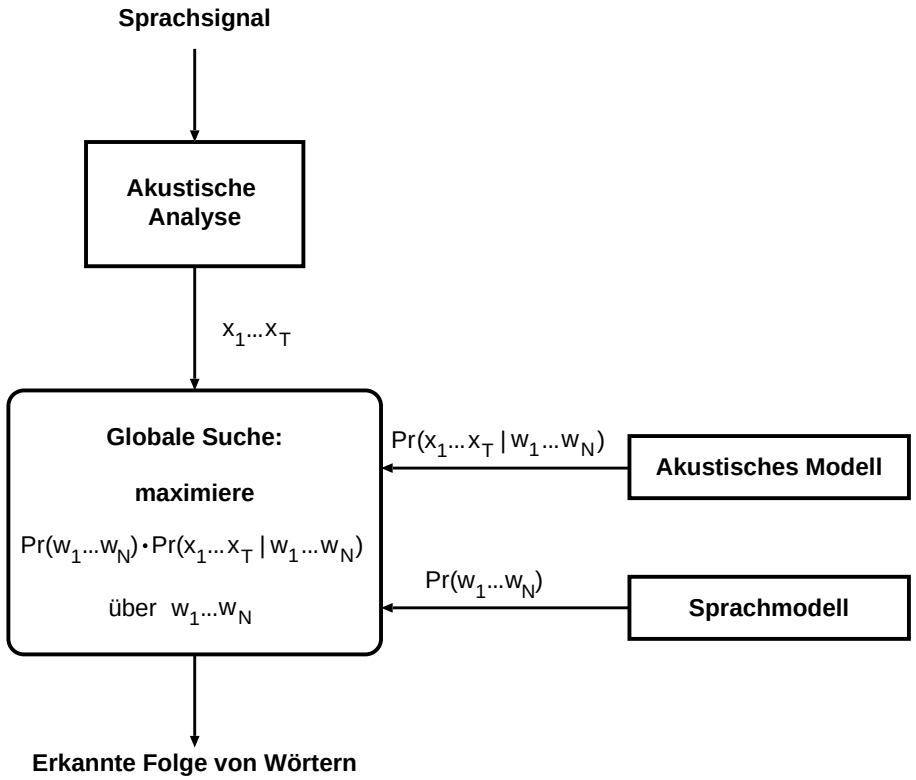

Abbildung 2.1: Komponenten des statistischen Spracherkenners.

gel liefert keinen Hinweis auf den strukturellen Aufbau eines Sprachmodells. Die Literatur kennt daher eine Vielzahl von unterschiedlichen Sprachmodellen, die bekanntesten sind in [Jelinek 91] beschrieben. Aufgrund der Kettenregel

$$\begin{aligned} p_\theta(w_1^N) &= \prod_{n=1}^{N} p_\theta(w_n|w_1^{n-1}) \\ &= \prod_{n=1}^{N} p_\theta(w_n|h_n) \end{aligned}$$

wird ein Sprachmodell häufig als bedingte Verteilung $p_\theta(w_n|h_n)$ formuliert, mit der die Wahrscheinlichkeit des Wortes w_n an der Stelle n der Wortkette w_1^N gegeben alle Vorgängerwörter w_1^{n-1} bestimmt wird. Für w_1^{n-1} wird auch kurz h_n als sogenannte Worthistorie geschrieben. Das häufigste Sprachmodell ist das m–gramm–Sprachmodell [Jelinek 91]. Die Abhängigkeit von den w_1^{n-1} Vorgängerwörtern wird hier auf die letzten $m-1$ Vorgängerwörter verkürzt:

$$p_\theta(w_n|h_n) = p_\theta(w_n|w_{n-m+1}^{n-1}) \quad ,$$

wobei die Wahrscheinlichkeiten unabhängig von der Position n in der Wortkette sind. Die Struktur dieses Sprachmodells basiert also auf Worttupeln der Länge m. Von den

führenden Forschungsgruppen werden zur Zeit hauptsächlich Trigramm–Sprachmodelle ($m = 3$) verwendet:

$$p_\theta(w_n|h_n) = p_\theta(w_n|w_{n-2}, w_{n-1}) \quad ,$$

kurz $p(w|h) = p(w|u, v)$, häufig um ausgewählte Viergramm– oder Fünfgramm–Tupel ergänzt [Sankar et al. 98, Seymore et al. 98, Woodland et al. 98]. Das Schema eines Trigramm–Sprachmodells findet sich in Abb. 2.2. Dieses Schema läßt sich auch als Markov–Kette 2. Ordnung oder als stochastische reguläre Grammatik interpretieren.

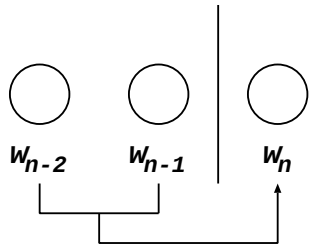

Abbildung 2.2: Schema eines Trigramm–Sprachmodells.

Nachdem die Struktur des Sprachmodells festgelegt ist, muß der Parametersatz θ trainiert werden. Dies geschieht mit Hilfe der bekannten Maximum–Likelihood–Schätzung [Lehmann 83]. Die Parameter des Sprachmodells werden dabei so gewählt, daß es auf einer gewählten, für die Spracherkennungsaufgabe typischen Wortkette w_1^N maximale Wahrscheinlichkeit erzielt. Damit wird diese Wortkette vom Sprachmodell optimal beschrieben. Formal geschieht dies über die Optimierung der Likelihood–Funktion $\mathcal{L}_{w_1^N}(\theta)$:

$$\begin{aligned} \theta &:= \operatorname*{argmax}_{\theta^*} \mathcal{L}_{w_1^N}(\theta^*) \\ \mathcal{L}_{w_1^N}(\theta^*) &:= p_{\theta^*}(w_1^N) \quad . \end{aligned}$$

Die verwendete Wortkette wird „Trainingskorpus“ genannt und kann mehrere hundert Millionen Wörter umfassen. In der Praxis benutzt man statt der Likelihood–Funktion die log–Likelihood–Funktion:

$$F_{w_1^N}(\theta^*) := \log \mathcal{L}_{w_1^N}(\theta^*) \quad .$$

Aufgrund der Monotonie des Logarithmus erreicht die log–Likelihood–Funktion für dieselben Parameter θ das Optimum wie die Likelihood–Funktion, läßt sich aber für die in dieser Arbeit vorgestellten Modelle analytisch leichter handhaben. Die log–Likelihood–Funktion kann nicht nur zur Schätzung von Parametern, sondern auch zur Auswahl von Strukturen in der Sprachmodellierung verwendet werden. Dies macht sie zu einem zentralen Konzept dieser Arbeit. Für das Trigramm–Sprachmodell ergibt sich die log–Likelihood–Funktion

$$\begin{aligned} F_{w_1^N}(\theta^*) &= \sum_{n=1}^{N} \log p_{\theta^*}(w_n|w_{n-2}, w_{n-1}) \\ &= \sum_{u,v,w} N(u, v, w) \cdot \log p_{\theta^*}(w|u, v) \end{aligned}$$

mit $N(u,v,w)$ als Häufigkeit des Trigramms (u,v,w) im Trainingskorpus. Unter Berücksichtigung der Normierungseigenschaft $\sum_{w \in W} p_\theta(w|h) = 1$ über alle Wörter w des Vokabulars W ergibt sich

$$p_{\theta^*}(w|u,v) = \frac{N(u,v,w)}{N(u,v)} \tag{2.1}$$

als Optimum der log–Likelihood–Funktion. Nach dem Ergebnis Gl. (2.1) bestehen die Parameter der Verteilung aus den einzelnen Wahrscheinlichkeiten selbst, weshalb auch von einem parameterfreien Modell gesprochen wird. Deshalb kann zur Vereinfachung der Schreibweise der Parametersatz θ^* weggelassen werden.

2.1.3 Bewertung von Sprachmodellen

Für die Bewertung eines Sprachmodells sind zwei Möglichkeiten üblich, die auch beide, wenn möglich, in dieser Arbeit verwendet werden:

- **Perplexität**
  Die Perplexität leitet sich aus der Entropie ab [Jelinek 91] und ist der Kehrwert des geometrischen Mittels der bedingten Sprachmodellwahrscheinlichkeiten über eine Wortkette ${w'}_1^{N'}$ der Länge N':

  $$PP := \frac{1}{\sqrt[N']{\prod_{n=1}^{N'} p_\theta({w'}_n|{h'}_n)}} \quad .$$

  Für eine faire Bewertung darf die Wortkette ${w'}_1^{N'}$ nicht im Trainingskorpus enthalten sein. In diesem Fall ist sie das „Testkorpus“, und die darauf berechnete Perplexität wird „Test–Set–Perplexität“ oder kurz „Testperplexität“ genannt. Analog heißt die Perplexität auf dem Trainingskorpus „Trainingsperplexität“. Für die Gleichverteilung $p_\theta({w'}_1^{N'}) = W^{-1}$ ergibt sich die Vokabulargröße W als Wert der Perplexität. Umgekehrt zeigt [Jelinek 91], daß für andere Sprachmodellverteilungen die Perplexität PP einer Gleichverteilung der Vokabulargröße PP über dem Testkorpus entspricht. Für den Suchprozeß interpretiert man daher die Perplexität PP als die mittlere Anzahl an Wörtern, aus denen er die gesuchte Wortfolge w_1^N zu bilden hat. Je geringer die Perplexität, desto enger die Auswahl.

  Weiter läßt sich die Perplexität

  $$\begin{aligned} PP &:= \frac{1}{\sqrt[N']{\prod_{n=1}^{N'} p_\theta({w'}_n|{h'}_n)}} \\ &= \frac{1}{\sqrt[N']{p_\theta({w'}_1^{N'})}} \\ &= \frac{1}{\sqrt[N']{\mathcal{L}_{{w'}_1^{N'}}(\theta)}} \end{aligned}$$

als längennormierte Likelihood auf den Testdaten interpretieren. Entstammen Trainings– und Testkorpus derselben Quelle, so verspricht die log–Likelihood–Schätzung der Sprachmodellparameter auf dem Trainingskorpus auch eine geringe Perplexität auf dem Testkorpus. Da log–Likelihood und Perplexität eines Modells auf den Trainingsdaten für denselben Parametersatz das Optimum erreichen, werden diese beiden Begriffe im Folgenden synonym verwendet.

- **Wortfehlerrate**
  Eine realistischere und praxisnähere Bewertung ist die Auswirkung des Sprachmodells auf die Wortfehlerrate. Die Wortfehlerrate ist die minimale Anzahl an Einfügungen („insertions“), Weglassungen („deletions“) und Ersetzungen („substitutions“), um von der tatsächlich gesprochenen auf die erkannte Wortfolge zu kommen, im Verhältnis zur Länge der tatsächlich gesprochenen Wortfolge. Die minimale Anzahl an Fehlern wird über eine Anpassung des Levenshtein–Abstandes [Levenshtein 66] für Wortfolgen bestimmt. Nachteile dieser Bewertung sind:

  – In die Bewertung geht auch die Qualität der akustischen Modellierung und der Steuerung des Suchprozesses ein.
  – Die Durchführung einer Erkennung ist deutlich aufwendiger als die Berechnung der Perplexität.

  Der Aufwand kann durch Wortgraphen [Ortmanns et al. 97] gemindert werden, die die aussichtsreichsten Zwischenschritte des Suchprozesses und die akustischen Bewertungen („Scores“) einer Erkennung festhalten. Durch das Sprachmodell erfolgt dann lediglich eine Neubewertung („Rescoring“) der Zwischenergebnisse. In der Erkennung kann daher ein einfaches Trigramm–Sprachmodell und im Rescoring, das der Erkennung als Post–Processing folgt, ein komplexeres Sprachmodell verwendet werden [Seymore et al. 98].

2.2 Erweiterung des Trigramm–Sprachmodells

In Kapitel 2.2 werden diejenigen Erweiterungen des in Kapitel 2.1 vorgestellten Trigramm–Sprachmodells beschrieben, die im Rahmen dieser Arbeit näher untersucht werden sollen. Die Reihenfolge der Erweiterungen ergibt sich aus dem Aufwand, gemessen an Laufzeit und Speicherplatzbedarf, der für die Verwendung der jeweiligen Erweiterungen im Sprachmodell erwartet wird. Dieser Aufwand und der damit erzielte Gewinn sollen am Schluß der Arbeit in Verhältnis zueinander gesetzt werden. Weiter wird in Kapitel 2.2 die bisher vorhandene Literatur zu den vorgestellten Erweiterungen diskutiert, und darin noch offene Fragen werden herausgestellt.

2.2.1 Glättungsverfahren

Eine typische Erkennungsaufgabe hat ein Vokabular von mehreren tausend Wörtern. Für z.B. $W = 20\,000$ Wörter ergeben sich somit $20\,000^3$ oder $8\,000\,000\,000\,000$ mögliche Trigramme. Ein übliches großes Trainingskorpus hat aber höchstens mehrere hundert

Millionen Wörter. Damit ist nur ein sehr kleiner Teil der möglichen Trigramme im Trainingskorpus beobachtbar. Nach Gl. (2.1) ergibt sich für nicht beobachtete Trigramme eine geschätzte Wahrscheinlichkeit von Null, und es können keine Wortketten erkannt werden, die solche Trigramme enthalten. Dies ist das sogenannte „Zero–Frequency–“ oder „Sparse–Data–Problem“.

Um solche starken Einschränkungen zu vermeiden, werden Glättungsverfahren eingesetzt. Bei solchen Verfahren wird ein Teil der den beobachteten Trigrammen zugewiesenen Wahrscheinlichkeit abgezogen, gesammelt und über die ungesehenen Trigramme verteilt. Dabei wird die Umverteilung der Wahrscheinlichkeitsmasse von der nächst gröberen Struktur abhängig gemacht:

$$\text{Trigramm } (u,v,w) \rightarrow \text{Bigramm } (v,w) \rightarrow \text{Unigramm } (w) \rightarrow \text{Zerogramm } ().$$

„Zerogramm“ bezeichnet üblicherweise die Gleichverteilung über die Wörter des Vokabulars. Der erste Ansatz für die Glättung war die lineare Interpolation der relativen Häufigkeiten [Bahl et al. 83]. Da es auf dem Trainingskorpus per Definition keine ungesehenen Trigramme gibt, müssen die Interpolationsgewichte über Kreuzvalidierung [Duda & Hart 73] bestimmt werden. Dazu wird das Trainingskorpus partitioniert und komplett durchlaufen. Die Interpolationsgewichte werden mittels EM–Algorithmus [Baum 72] an jeder Korpusposition ermittelt, wobei zur Bestimmung der relativen Häufigkeiten die Partition, in der man sich gerade befindet, herausgenommen („deleted“) wird und die relativen Häufigkeiten somit über die verbleibenden Trigramme bestimmt werden. Trigramme, die nur in der herausgenommenen Partition vorkommen, werden dadurch zu ungesehenen Ereignissen. Diese Methode wurde als „deleted interpolation“ erstmals allgemein in [Jelinek & Mercer 80] vorgestellt.

Die erste Alternative ergab sich durch die Verwendung des „Katz–Discounting“ [Katz 87] in der Sprachmodellierung [Jelinek 91]. Die relativen Häufigkeiten wurden zwar noch multiplikativ gewichtet, aber nicht mehr linear interpoliert, sondern entsprechend der Hierarchie wurde die feinste Verteilung, die Trigramm–Verteilung, ausgewählt (Backing–Off–Glättung). Nur für den Fall ungesehener Trigramme wurden mit einer Restwahrscheinlichkeit die Bigramme herangezogen, nur für ungesehene Bigramme die Unigramme und nur für ungesehene Unigramme das Zerogramm. Die Schätzung der Parameter erfolgte nicht durch Kreuzvalidierung, sondern durch die „Turing–Good–Zähler“ [Good 53].

In [Ney et al. 95] wurde beobachtet, daß die nach Turing–Good geglätteten Häufigkeiten einen nahezu konstanten Abstand zu den reinen relativen Häufigkeiten haben. Dies wurde als eine Motivation für das bereits kurz vorher vorgeschlagene absolute Discounting [Ney et al. 94] angegeben, bei dem die Zähler der relativen Häufigkeiten um einen konstanten Discounting–Wert vermindert wurden. Dieser Wert wurde durch eine Leaving–One–Out–Schätzung bestimmt, eine Form der Kreuzvalidierung, bei der die herausgenommenen Partitionen die Größe 1 haben, also nur aus dem gerade beobachteten Ereignis bestehen. Für diesen Spezialfall läßt sich eine geschlossene Lösung sowohl für den Discounting–Wert als auch für die Gewichte der linearen Interpolation [Ney et al. 94] approximieren. Auf dem LOB–Korpus mit einer Länge von 1,1 Millionen Wörtern erzielte das absolute im Vergleich zum linearen Discounting eine Reduktion der Bigramm–Testperplexität um 11% (577→514) bei einem Vokabular mit 50 000 Wörtern. Die durch

Leaving–One–Out geschätzten Parameter lagen dicht bei den optimalen Parametern auf dem Testkorpus.

Durch Leaving–One–Out läßt sich auch eine Alternative zu den relativen Häufigkeiten als Glättungswahrscheinlichkeiten (Bi- und Unigramme) durch die sogenannte Singleton–Verteilung herleiten, die eine Statistik über einmal gesehene Ereignisse darstellt [Kneser & Ney 95]. Diese Singleton–Verteilung erbrachte zusammen mit Absolute Discounting und Backing–Off auf einem 40–Millionen–Wort–Auszug des Wall–Street–Journal–Korpus bei einem Vokabular von 45 000 Wörtern eine Verringerung der Testperplexität um 10% (161.0→145.7) und eine Verringerung der Wortfehlerrate um 6% (11.9%→11.2%) relativ.

Diskussion. Die vorgestellten Ergebnisse legen nahe, daß das absolute Discounting mit Backing–Off und Singleton–Glättung hinsichtlich der Perplexität das beste Verfahren ist. Es fehlt jedoch ein systematischer Vergleich aller Glättungsverfahren auf einem großen Textkorpus. Mit Ausnahme von [Kneser & Ney 95] ist auch nicht untersucht worden, ob sich die Reduktionen in der Perplexität auch ebenso deutlich auf die Fehlerrate auswirken.

2.2.2 Varigramme

Variable m–gramme, kurz Varigramme, stellen eine Erweiterung der Trigramme dar. Das Trigramm–Sprachmodell wird um einzelne m–gramme ($m > 3$) ergänzt, von denen angenommen wird, daß sie die Vorhersage eines aktuellen Wortes w verbessern können (Abb. 2.3). Z. B. ist das Trigramm $p(w|$„NEW“„YORK“) eigentlich nur ein Bigramm, weil der Städtename „New York“ in zwei Wörter zerfällt. Hier ist die Erweiterung um das Vorgängerwort zu „New York“ hilfreich.

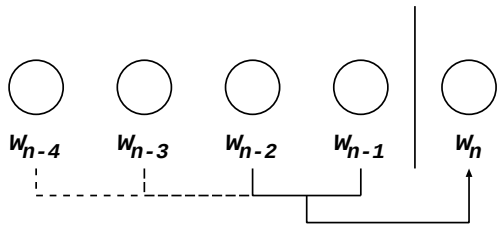

Abbildung 2.3: Schema eines Varigramm–Sprachmodells.

In [Ron et al. 94] wird für Buchstaben statt Wörter der sogenannte „Prediction Suffix Tree“ vorgestellt. Dies ist ein Baum, dessen Knoten mit je einem Buchstaben markiert sind, so daß sich von jedem Knoten bis zur Wurzel eine Buchstabenkette zusammensetzen läßt. Weiter ist jedem Knoten eine bedingte Wahrscheinlichkeitsverteilung über die Vorgängerbuchstaben zugeordnet, so daß durch Aufmultiplizieren dieser bedingten Wahrscheinlichkeiten beim Durchlauf von der Wurzel zu einem Knoten jeder so erzeugbaren Buchstabenkette ihre Verbundwahrscheinlichkeit zuzuordnen ist, wobei sich die

Summe über alle Ketten gleicher Länge zu eins ergibt. Als Wahrscheinlichkeiten werden nach Maximum Likelihood die relativen Häufigkeiten über einer Menge von Trainingsketten gewählt. Nach Maximum Likelihood wird auch die Struktur des Baumes bestimmt: Ausgehend vom Wurzelknoten mit der Unigrammverteilung über die Buchstaben wird der Baum Ebene für Ebene um jene Blätter erweitert, deren bedingte Verteilungen einen Gewinn in der log–Likelihood erwirken, der über einem vorher festgelegtem Schwellwert liegt. In einer unveröffentlichten Arbeit [Pereira et al. 96] ist das Konzept auf die Sprachmodellierung übertragen und zu Mixtures von Prediction Suffix Tree erweitert worden. Im Vergleich zum Trigramm wurde damit eine Verbesserung in der Testperplexität auf einem Auszug des North American Business Corpus mit einer Länge von 32,5 Millionen Wörtern um bis zu 15% (247.7→210.6) erzielt. Der Effekt der Mixtures wurde dabei aber nicht vom Effekt der Varigramme getrennt.

In [Niesler & Woodland 96] wurde ein sehr ähnliches Konzept als Teil eines wortkategoriebasierten Sprachmodells eingeführt. Die Wahrscheinlichkeit der Kategorie des vorherzusagenden Wortes wurde anhand der vorangehenden Kategorien unterschiedlicher Länge bestimmt. Diese wurden als Baum mit der Unigrammverteilung an der Wurzel abgelegt. An den Knoten findet sich je eine Wortkategorie und diejenige bedingte Wahrscheinlichkeit, die sich für die Historie aus den Wortkategorien des Pfades von der Wurzel bis zu diesem Knoten ergibt. Die Baumstruktur wurde ähnlich wie für die Prediction Suffix Trees erzeugt, indem, ausgehend von der Wurzel, Ebene für Ebene die vorhandenen Blätter um eine Vorgängerkategorie erweitert wurden, falls der Gewinn in log–Likelihood auf den Trainingsdaten über einem gegebenen Schwellwert liegt. Es wurde hier die Leaving–One–Out–log–Likelihood verwendet, um ungesehene Ereignisse zu simulieren und dadurch nur robuste Historien auszuwählen. Dieses wortkategoriebasierte Sprachmodell gewinnt auf dem LOB–Korpus gegenüber dem wortkategoriebasierten Trigramm 2% (544.1→534.1), verliert jedoch gegenüber dem wortbasierten Trigramm 13% (474.0→534.1) in der Testperplexität. Auf dem Switchboard–Korpus mit einer Länge von 2 Millionen Wörtern und einem Vokabular mit 23 000 Wörtern verschlechtert es sich sogar um 49% (92.94→138.53), allerdings bei einer Anzahl von Parametern von 5% (1 201 176→54 547) der Zahl der Trigramm-Parameter. Auch hier wurden die Effekte der Wortkategorisierung und des Varigramms nicht getrennt. Wie später zu sehen sein wird, liegen die Perplexitäten für wortkategoriebasierte Sprachmodelle meistens über denen der wortbasierten.

[Kneser 96] verwendet einen anderen Ansatz, in dem nicht die Historien mit zugehörigen bedingten Verteilungen, sondern die einzelnen m–gramme in einem Baum abgelegt werden. Anstatt einen Baum um geeignete m–gramme zu erweitern, werden alle m–gramme einer vorgegebenen maximalen Länge zuerst zugelassen, um anschließend ein „Pruning“ durchzuführen. Bei einem Pruning werden diejenigen m-gramme gelöscht, die die Trainingsperplexität am wenigsten verschlechtern, bis eine vorgegebene Zahl an Parametern erreicht ist. Für dieses Sprachmodell verringert sich die Testperplexität auf dem North American Business Corpus für ein gepruntes Fünfgramm im Vergleich zu einem geprunten Trigramm mit derselben Parameterzahl um 9% (140.3→127.9). Die angegebenen Fehlerraten für ein gepruntes Viergramm im Vergleich zu einem geprunten Trigramm mit derselben Parameterzahl verringern sich unmerklich (10.9%→10.8%). Diese Technik

bzw. ihre Vereinfachung, nur m–gramme ab einer bestimmten Häufigkeit zuzulassen, findet sich in einigen aktuellen Sprachmodellen [Seymore et al. 97][Klakow et al. 98].

Diskussion. Aus der Literatur ist ein klarer positiver Effekt der Varigramme zu erkennen. Leider ist in den ersten beiden Quellen dieser Effekt nicht von dem anderer Ansätze getrennt worden. Fehlerraten sind nur in der dritten Quelle angegeben. Diese sind im Vergleich zur Verringerung in der Perplexität enttäuschend. Es bleibt offen, ob diese Beobachtung generell für Varigramme gilt oder nur durch die andere Auswahlmethode bedingt ist.

2.2.3 Wortphrasen

Wortphrasen sind ein alternativer Ansatz zu den Varigrammen für die Kontexterweiterung. Anstatt die Historienlänge zu erweitern, werden geeignete Wortpaare (oder auch längere Wortfolgen) wie z.B. „NEW YORK“ als neues Wort (Wortphrase) „NEW_YORK“ definiert und das Vokabular um diese neuen Wortphrasen erweitert (Abb. 2.4). Der Vorteil ist, daß nun wirklich jedesmal, wenn „NEW YORK“ im Korpus erscheint, der Kontext erweitert wird, und daß es sich formal immer noch um ein Trigramm handelt, wenn auch über einem erweiterten Vokabular. Der Nachteil ist, daß sich Wörter und Phrasen nicht eindeutig zuordnen lassen: Auf eine Wortfolge können eventuell zwei überlappende Phrasen passen, und die Bestandteile der Phrasen sind nach wie vor Wörter des Vokabulars mit positiver Wahrscheinlichkeit. Die Wahrscheinlichkeit einer Wortkette ist dann die Summe der Wahrscheinlichkeiten aller auf ihr bildbaren Phrasenketten (Summenkriterium) oder angenähert die Wahrscheinlichkeit der wahrscheinlichsten Phrasenkette (Maximum–Approximation).

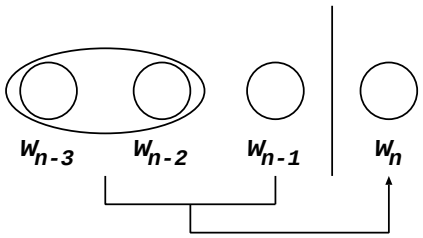

Abbildung 2.4: Schema eines Wortphrasen–Sprachmodells.

In [Brown et al. 92] werden die Wortpaare, sogenannte „Sticky Pairs“, durch ein vereinfachtes Mutual–Information–Kriterium, nämlich dem logarithmierten Verhältnis der Bigramm– zum Produkt der beiden Unigramm-Wahrscheinlichkeiten, bestimmt. Beispiele für gefundene Wortpaare werden angegeben, jedoch keine Modelle oder Perplexitäten.

In [Giachin et al. 94] wurde ein klassenbasiertes Bigramm–Sprachmodell durch Zusammenlegen jener Wortklassenpaare verbessert, die die Trainingsperplexität, also die log–Likelihood auf den Trainingsdaten, am meisten verringern und deren Häufigkeit über

einem gegebenen Schwellwert liegt. Die gewählten Klassenpaare nahmen an den folgenden Auswahlen teil, so daß zusammengesetzte Phrasen entstehen konnten. Für 199 Klassen konnte im Vergleich zum ursprünglichen Bigramm–Modell eine Verbesserung von 22% (34.2→26.7), zum Trigramm–Modell um 4% (27.9→26.7) in der Testperplexität auf einem Fahrplanauskunfts–Korpus mit einer Länge von ca. 60 000 Wörtern und einem Vokabular mit ca. 800 Wörtern erzielt werden. Die Auswahl anhand der Leaving–One–Out–Trainingsperplexität sowie ein Pruning gewählter Phrasen brachten keine weitere Verbesserung. In [Giachin 95] wird auf demselben Korpus dieses Verfahren für 150 Klassen mit dem in [Brown et al. 92] angegebenen verglichen. Dabei ergab sich kein nennenswerter Unterschied. Bei den erstmals angegebenen Wortfehlerraten ergab sich für beide Kriterien eine relative Verbesserung von ca. 9% (17.4%→15.8%).

Auch [Ries et al. 96] verwendet die Leaving–One–Out Bigramm–Trainingsperplexität zur Phrasenauswahl für ein wort– und ein wortklassenbasiertes Trigramm–Sprachmodell und erzielt damit auf dem Switchboard–Korpus mit einer Länge von 2 Millionen Wörtern eine Reduktion in der Testperplexität von 2% (∼79.5→∼78) bzw. 1% (∼79→∼78) und auf dem Verbmobil–Korpus mit einer Länge von 300 000 Wörtern von 2% (∼68.8→∼67.5) bzw. 4% (∼66→∼63.3) im Vergleich zum entsprechenden phrasenlosen Trigramm–Sprachmodell. Eine ebenfalls nur geringe Verminderung in der Testperplexität auf einem Spontansprachenkorpus mit einer Länge von 200 000 Wörtern und einem Vokabular von 3 600 Wörtern berichten [Riccardi et al. 97] durch nach dem Trainingsperplexitäts–Kriterium ausgewählte Phrasen.

[Klakow 98a] wählt zusammengesetzte Phrasenpaare nach den Kriterien Mutual Information, Unigramm–log–Likelihood und Häufigkeit aus. Auf dem LDT–Korpus, einer Diktieraufgabe für juristische Texte mit einer Länge von 52 000 Wörtern, erbringen alle drei eine Verringerung in der Testperplexität um 17% (∼88→∼73) für Bigramme und um 2% (∼51.5→∼50.5) für Trigramme, vermutlich wegen des sehr kleinen Korpus. Es wurden also keine wesentlichen Unterschiede in der Performanz der Auswahlverfahren festgestellt, so daß [Klakow et al. 98], ähnlich wie [Gauvain et al. 97], die zusammengesetzten Phrasen rein nach der Häufigkeit auf dem Broadcast–News–Korpus mit einer Länge von 140 Millionen Wörtern bei einem ursprünglichen Vokabular von 64 000 Wörtern auswählt und damit eine Verringerung in der Trigramm–Testperplexität um 4% (180→172.7) erzielt. Mit einer ähnlichen, allerdings nicht näher beschriebenen Methode werden auf dem Wall–Street–Journal–Korpus mit einer Länge von 40 Millionen Wörtern und einem Vokabular von 5000 Wörtern die Trigramm–Testperplexität um 8% (60.8→55.8) und die Wortfehlerrate sogar um 13% relativ (7.0%→6.1%) verringert. Allerdings findet hier auch ein erneutes Training des akustischen Modells mit den Wortphrasen statt, das einen deutlichen Beitrag zur Verminderung der Wortfehlerrate leisten muß, da die Wortfehlerrate deutlich mehr als die Perplexität absinkt.

[Bimbot et al. 95] verfolgt einen anderen Ansatz, indem hier anfangs alle Phrasen bis zu einer vorgegebenen maximalen Länge mit einer initialen Wahrscheinlichkeit zugelassen sind. In einem Viterbi–Training der Phrasenwahrscheinlichkeiten wird das Trainingskorpus so in die Phrasen aufgespalten, daß die Unigramm–Trainingsperplexität in der Maximum–Approximation minimal wird. Die Phrasenwahrscheinlichkeiten werden über relative Häufigkeiten neu geschätzt, unterbewertete Phrasen geprunt und das Prozedere

wiederholt. Das so bestimmte Phrasen–Unigramm–Sprachmodell erzielte auf dem ATIS–Korpus mit einer Länge von 100 000 Wörtern und einem Vokabular 900 Wörtern eine Reduktion der Testperplexität um 16% (21.0→17.7) im Vergleich zum Wortbigramm als bestem Wort–m–gramm auf diesem Korpus. In [Deligne & Bimbot 95] wurden die Maximum–Approximation mit dem Summenkriterium und entsprechend das Viterbi– mit dem EM–Training verglichen mit dem Ergebnis, daß sich keine Verbesserung durch das Summenkriterium ergibt.

Darüber hinaus, und in dieser Arbeit nicht weiter diskutiert, ist der phrasenbasierte Sprachmodellansatz weiter entwickelt worden: [Hwang 97] und [Klakow 98a] versuchen eine Vokabularoptimierung mit Hilfe der vorgestellten Methoden, um die Testperplexität zu senken bzw. Out–of–Vocabulary–Wörter aus kleineren linguistischen Einheiten zusammenzusetzen. [McCandless & Glass 93, McCandless & Glass 94] benutzen die log–Likelihood–Phrasenauswahl zum Aufbau einer stochastischen Chunk–Grammatik. [Masataki & Sagisaka 96] bilden mit dem log–Likelihood–Kriterium in einem hybriden wort– und klassenbasierten Sprachmodell Wortphrasen heraus.

Diskussion. In der Literatur wird der Schwerpunkt auf das Auswahlverfahren gelegt. Dabei hat sich keines der untersuchten Verfahren als den anderen überlegen herausgestellt. Außer in [Bimbot et al. 95, Deligne & Bimbot 95] wird die Problematik von Summenkriterium und Maximum–Approximation nicht erwähnt, und in [Deligne & Bimbot 95] macht die Wahl der Methode keinen Unterschied. Die in [Giachin 95, Klakow et al. 98] angegebenen Fehlerraten sind erfolgversprechend. Die Reduktionen in der Perplexität schwanken jedoch sehr stark von Quelle zu Quelle, vermutlich weil vor allem in den frühen Arbeiten mit kleinen Korpora gearbeitet wurde.

2.2.4 Wortklassen

Haben die beiden bisher beschriebenen Erweiterungen das Trigramm–Modell verfeinert, so dienen die Wortklassen der Glättung der Trigramm–Wahrscheinlichkeiten. Wie bei den Glättungsverfahren beschrieben, gibt es bei einem Vokabular von 20 000 Wörtern $20\,000^3 = 8 \cdot 10^{12}$ mögliche Trigramme, deren Auftrittswahrscheinlichkeit mit den existierenden Korpora niemals gut zu schätzen sein wird. Teilt man hingegen das Vokabular in z. B. 100 Wortklassen auf, so gibt es nur noch $100^3 = 1\,000\,000$ mögliche Klassentrigramme, die sich sehr viel besser schätzen lassen (Abb. 2.5). Natürlich bedeutet diese Reduktion der Parameter eine drastische Vergröberung des Sprachmodells, die der Verbesserung des Sprachmodells durch eine bessere Parameterschätzung entgegenwirkt. Die Wortklassen können entweder einer innerhalb der Linguistik vorgegebenen Wortartenklassifikation entsprechen, sogenannte „Parts–of–Speech“ (POS), oder mit einem statistischen Verfahren, einem sogenannten Clusteralgorithmus, bestimmt werden. Diese automatische Aufteilung des Vokabulars ist ein kombinatorisches Problem, zu dessen Lösung bisher kein optimales Verfahren bekannt ist.

Als ältesten Ansatz adaptieren [Jelinek 91, Brown et al. 92] das bottom–up Clusterverfahren [Duda & Hart 73, pp. 230 und 235] auf die Wortklassenbildung. Anfangs hat

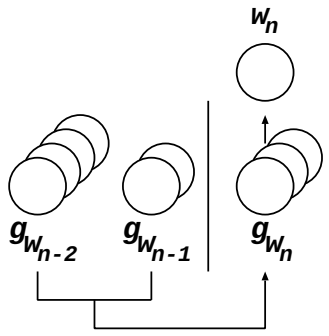

Abbildung 2.5: Schema eines wortklassenbasierten Sprachmodells.

jedes Wort seine eigene Klasse. Im Verlauf des Algorithmus werden in jedem Schritt diejenigen beiden Klassen verschmolzen, deren Zusammenlegung die Bigramm–Trainings–log–Likelihood am wenigsten verringern. Sobald die vorgegebene Klassenzahl erreicht ist, wird jedes Wort des Vokabulars in jede Klasse geschoben und verbleibt in derjenigen Klasse, in der es die größte log–Likelihood erzielt. Dieser Algorithmus wurde in [Brown et al. 92] auf einem Trainingskorpus mit einer Länge von 350 000 000 Wörtern und einem Vokabular von 250 000 Wörtern für 1000 Wortklassen angewandt. Das resultierende klassenbasierte Trigramm–Modell erzielte auf dem Brown–Korpus mit einer Länge von 1 Million Wörtern als Testkorpus nur eine Vergrößerung der Testperplexität um 11% (244→271) im Vergleich zum wortbasierten Trigramm–Modell. Erst die lineare Interpolation beider mit auf einem dritten Korpus trainiertem Interpolationsparameter erbrachte eine Reduktion der Testperplexität um 3% (244→236).

Die in [Brown et al. 92] anklingende Idee des Verschiebens der Worte des Vokabulars ist in [Kneser & Ney 93, Ney et al. 94] aufgegriffen worden. Hier handelt es sich um einen reinen Austausch–Algorithmus [Duda & Hart 73, pp. 227–228]. Das Vokabular wird willkürlich in die vorgegebene Anzahl Klassen aufgeteilt. In mehreren Durchläufen (Iterationen) wird nacheinander jedes Wort des Vokabulars in jede Klasse geschoben und verbleibt in derjenigen Klasse, in der es die größte Bigramm–log–Likelihood erzielt. Alternativ wird in [Kneser & Ney 93] auch die Leaving–One–Out Bigramm–log–Likelihood als Kriterium verwendet. Auf dem LOB–Korpus mit einer Länge von 1 Million Wörtern und einem Vokabular mit 50 000 Wörtern erzielten beide Kriterien eine Reduktion der Bigramm–Testperplexität um 12% (541→478) im Vergleich zum wortbasierten Bigramm–Modell, die Interpolation beider Modelle erbrachte eine Reduktion um 19% (541→439). Im Vergleich zu einem POS–Bigramm–Modell liegen diese Größen jeweils bei 14% (556→478) und 21% (556→439), die Interpolation aller drei Modelle erbrachte im Vergleich zum wortbasierten Bigramm eine Reduktion von 22% (541→420) und zum POS–Bigramm von 24% (556→420).

In [Jardino & Adda 93, Jardino 96] wird der Austausch–Algorithmus in den Metropolis–Algorithmus [Metropolis et al. 53] verändert: Ein Wort des Vokabulars und die Klasse, in die es verschoben werden soll, werden zufällig ausgewählt. Die Verschiebung wird auch bei einer Verschlechterung durchgeführt, wenn diese Verschlechterung der Bigramm–log–

Likelihood innerhalb eines mit der Zeit kleiner werdenden Bereiches liegt. Bei geeigneter Wahl der Steuerparameter dieses als Simulated Annealing bekannten Verfahrens konvergiert es im Gegensatz zu den beiden oben genannten zum globalen Optimum. [Jardino 96] erreicht auf einem Auszug des Wall–Street–Journal–Corpus von 30 Millionen Wörtern bei einem Vokabular von 20 000 Wörtern eine Verringerung der Perplexität um 23% (111→86) für das Klassentrigramm–Modell im Vergleich zum Worttrigramm–Modell. Eine weitere Verbesserung wird in [Jardino & Adda 94] durch die Zugehörigkeit eines Wortes zu mehreren Kategorien erzielt.

Auch [Niesler 97] läßt mehrere Kategorien pro Wort zu. Die Interpolation dieses wortkategoriebasierten Modells mit dem Worttrigramm erbrachte eine Verringerung um 11% (204.3→181.7) in der Perplexität und um 3% relativ (11.84%→11.44%) in der Wortfehlerrate auf dem Wall–Street–Journal mit einer Länge von 40 Millionen Wörtern als Trainings– und dem North–American–Business–Corpus 7000–Wort–H1–Development–Set als Testkorpus mit einem Vokabular von 65 000 Wörtern.

Diskussion. Die klassenbasierten Sprachmodelle zeigen einen deutlichen positiven Einfluß auf die Perplexität, allerdings bis auf [Kneser & Ney 93] nur in Interpolation mit dem wortbasierten Trigramm–Modell. Dabei beziehen sich die vorgestellten Cluster–Verfahren alle nur auf ein Bigramm–Kriterium, obwohl in den Versuchen Klassentrigramm–Modelle verwendet werden. Vermutlicher Grund ist die hohe Laufzeitkomplexität des Clusterverfahrens, da es sich um ein kombinatorisches Problem handelt, allerdings wird außer in [Brown et al. 92] auf dieses Problem nicht oder nur grob eingegangen. Weiter werden in der Literatur außer in [Niesler 97] für wortklassenbasierte Sprachmodelle keine Fehlerraten angegeben.

2.2.5 Abstand–m–gramme

Abstand–m–gramme sind m–gramme über nicht aufeinanderfolgende Korpuspositionen. Die Abstand–2–Trigramme z. B. sind die Randsummen des Viergramms über die zweite bzw. dritte Wortposition. Als Ergänzung zum normalen Trigramm dienen die Abstand–Trigramme zur Kontexterweiterung (Abb. 2.6).

Es ist naheliegend, diese drei Trigramm–Modelle linear zu interpolieren, jedoch ist keine vollständige Untersuchung dazu bekannt. In [Rosenfeld 94] werden Abstand–m–gramme im Kontext des Maximum–Entropy–Konzeptes [Jaynes 57, Berger et al. 96] untersucht. In diesem Ansatz zur Kombination verschiedener Wissensquellen wird das Modell lediglich über die Randsummen seiner Verteilung (sogenannte Constraint–Gleichungen) definiert, es wird sonst keine Annahme getroffen (maximale Entropie). [Berger et al. 96] weisen nach, daß daraus ein log–lineares Modell resultiert, das der aus der statistischen Mechanik bekannten Gibbs–Verteilung [Landau & Lifschitz 87, Kap. III][Binney et al. 92, pp. 34–35] entspricht. Umgekehrt sind die Constraint–Gleichungen das Ergebnis der Maximum–Likelihood–Parameterschätzung für die Gibbs–Verteilung. [Rosenfeld 94] berichtet für ein Maximum–Entropy–Sprachmodell mit Abstand–2–Trigrammen und –Bigrammen, das auf einem Auszug des Wall–Street–Journal–Korpus mit 1 Million

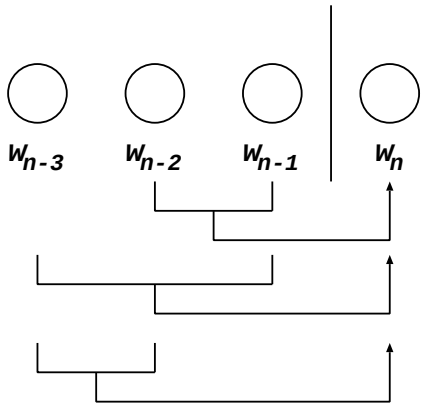

Abbildung 2.6: Schema eines um Abstand–2–Trigramme erweiterten Trigramm–Sprachmodells.

Wörtern bei einem Vokabular mit 20 000 Wörtern trainiert wurde, eine Verminderung in der Testperplexität um 7% (269→249) im Vergleich zu einem Standard–Trigramm–Sprachmodell auf einem weiteren Auszug mit 325 000 Wörtern. [Rosenfeld 94] erwähnt auch die Verknüpfung von Abstand–m–Bigrammen mittels linearer Interpolation. Untersuchungen dazu stellt er jedoch nicht vor.

Diskussion. Für Abstand–m–gramme gibt es nur Ergebnisse für die Maximum–Entropy–Methode, nicht für die lineare Interpolation. Für den Maximum–Entropy–Ansatz gibt es auch nur Perplexitäten auf dem kleinen Auszug mit einer Million Wörtern, die jedoch vielversprechend sind. Es werden keine Fehlerraten angegeben.

2.2.6 Andere Erweiterungen

Es gibt noch weitere, im Verlauf dieser Arbeit nicht weiter berücksichtigte Ergänzungen oder Alternativen zum Trigramm–Sprachmodell.

Dynamische Sprachmodelle. Bei den bisherigen Ansätzen wurden die Wahrscheinlichkeiten über einem Trainingskorpus ermittelt und blieben danach fest. Bei den dynamischen Modellen werden diese Wahrscheinlichkeiten in Richtung der bisher im Testkorpus beobachteten Wörter abgeändert. Das einfachste Modell ist ein Cache [Jelinek et al. 91, Generet et al. 95], bei dem eine Unigramm– oder Bigramm–Statistik über die bisherigen Ereignisse im Testkorpus geführt und zum Trigramm–Modell hinzuinterpoliert wird. Eine Erweiterung sind die Trigger [Rosenfeld 94, Tillmann & Ney 96, Tillmann & Ney 97], geeignete Wortpaare, die über beliebig lange Distanzen im Trainingskorpus stehen und die Eigenschaft haben, daß nach der Beobachtung des ersten Wortes das zweite mit erhöhter Wahrscheinlichkeit auftritt, z. B. „AIRLINES“→„PASSENGERS“. Auch diese Paare werden nach dem Likelihood–Kriterium auf dem Trai-

ningskorpus ausgewählt und ihre Wahrscheinlichkeiten geschätzt. Andere Arbeiten [Iyer & Ostendorf 96, Kneser & Peters 97] teilen das Trainingskorpus auf und trainieren auf diesen Teilen separate m–gramm–Sprachmodelle. Diese Teilmodelle werden interpoliert und je nach ihrer Perplexität auf der bisherigen Worthistorie im Testkorpus unterschiedlich gewichtet. Die Aufteilung des Trainingskorpus kann entweder automatisch ähnlich der Wortklassenbildung geschehen oder themenspezifisch per Hand.

Stochastische Grammatiken. Es gibt Ansätze, auf den Traingskorpora stochastische kontextfreie Grammatiken zu trainieren [Jelinek et al. 90, Ney 92], deren Regeln entweder linguistisch bestimmt oder aber automatisch nach einem statistischen Kriterium ausgewählt werden. Diese Grammatik kann dann in einem CYK–Parsing oder einem analogem Parsingverfahren verwendet werden. Da kontextfreie Regeln, die natürlichsprachliche Sätze vollständig abdecken, nur schwer zu finden sind, gibt es auch Teillösungen. Beim Chunk–Parsing [McCandless & Glass 93, McCandless & Glass 94] z. B. decken die Regeln nur kürzere Wortfolgen ab, die dann wiederum in einem geglättetem m–gramm–Modell aufgehen. Eine andere Alternative sind die „Link Grammars“, [Della Pietra et al. 94, Collins 96, Collins 97], auch als „Dependency Grammars“ bezeichnet, die die unmittelbaren Zusammenhänge (Links) unter den Wörtern beschreiben. Die Link Grammars sind in jüngeren Arbeiten bereits für die Sprachmodellierung verwendet worden [Chelba et al. 97, Chelba & Jelinek 98].

2.3 Problemstellung

Aus der Diskussion der Literatur in Kapitel 2.2 ergeben sich für diese Arbeit die folgenden Schwerpunkte, die hier nach dem Aufwand für die daraus resultierenden Sprachmodelle aufgelistet sind:

1. **Glättungsverfahren**

   Es soll ein systematischer Vergleich der aus der Literatur bekannten Glättungsverfahren auf großen Texten für Perplexität und Wortfehlerrate durchgeführt werden. Damit soll untersucht werden, ob eine dieser Methoden durchgängig die besten Ergebnisse liefert und somit als Standardmethode für die folgenden Arbeiten verwendet werden kann.

2. **Varigramme**

   Das Sprachmodell soll wie in [Ron et al. 94, Niesler & Woodland 96] immer um ganze bedingte Verteilungen, also historienbasiert, anstatt um einzelne Varigramme wie in [Kneser 96] erweitert werden. Damit soll einmal der Effekt der Varigramme für diese Auswahlmethode von anderen Methoden getrennt werden, was in der Literatur nicht gemacht wird. Auch soll untersucht werden, ob die in [Kneser 96] beobachtete Diskrepanz in der Verringerung der Perplexität und Fehlerrate auch mit dieser Auswahlmethode bestehen bleibt und somit der Varigramm–Ansatz zukünftig verwendet werden kann.

3. Wortphrasen

Zu Beginn dieser Arbeit lagen nur Ergebnisse auf kleinen Korpora vor, die durch starke Schwankungen in der Verminderung der Perplexität gekennzeichnet waren. Es soll daher auf großen Korpora der Effekt der Wortphrasen untersucht werden, insbesondere, ob das Auswahlverfahren tatsächlich kaum Einfluß auf die Perplexität hat und auch ob das Summenkriterium nicht durch einfachere Verfahren, z.B. die Maximum–Approximation, ersetzt werden kann. Auch soll untersucht werden, ob, anders als in der Literatur für die Varigramme beschrieben, die Wortphrasen auch einen positiven Effekt auf die Fehlerrate haben und somit die bessere Alternative für die Kontexterweiterung sind.

4. Wortklassen

In der Literatur wird stets ein Bigramm–Kriterium für den Clusteralgorithmus verwendet, obwohl im Sprachmodell ein klassenbasiertes Trigramm benutzt wird. Es soll daher geprüft werden, ob die Verwendung eines Trigramm–Kriteriums möglich im Sinne vertretbarer Laufzeiten ist, welche Verfeinerungen am Algorithmus dazu notwendig sind und ob sich das Kriterium in Form geringerer Perplexitäten auszahlt. Da alle Quellen das Austausch–Verfahren entweder zur Abrundung [Brown et al. 92] oder als eigentlichen Algorithmus [Kneser & Ney 93, Jardino & Adda 93] einsetzen, soll es auch in dieser Arbeit verwendet werden. Eine Erweiterung dieses Algorithmus um Simulated Annealing oder mehrere Klassen pro Wort ist nicht vorgesehen, da es nicht zu erwarten ist, daß der dazu notwendige deutliche Mehraufwand zu entsprechend besseren Ergebnissen führen wird. Weiter soll die Auswirkung klassenbasierter Sprachmodelle auf die Fehlerrate untersucht werden.

5. Abstand–Trigramme

Für interpolierte Abstand–m–gramme müssen Perplexitäten und Fehlerraten bestimmt und damit geprüft werden, ob dieser Ansatz hinreichend Verminderungen bringt. Da hier im Gegensatz zu den bisherigen Erweiterungen drei nahezu gleichberechtigte Sprachmodelle integriert werden, soll auch Maximum Entropy als Alternative zur linearen Interpolation untersucht werden.

6. Gemeinsame Verwendung der ausgewählten Wortabhängigkeiten

Die vorgestellten Erweiterungen des Trigramms wurden bisher nur separat, zum Teil auf kleineren Texten und ohne Fehlerraten getestet. Es sollen daher systematische Tests auf großen Daten durchgeführt werden, die den kumulativen Gewinn beim gemeinsamen Einsatz dieser Erweiterungen messen. Dies gilt sowohl für die Perplexität als auch für die Fehlerrate. Diesr Gewinn soll in Verhältnis zum Aufwand der Verwendung der jeweiligen Erweiterungen, gemessen in Speicherplatzbedarf und Laufzeit, gesetzt werden.

Alle Untersuchungen wurden an drei Korpora durchgeführt. Genauere Beschreibungen dieser Korpora finden sich in Anhang A.

- **WSJ0:** Das Kürzel steht für „Wall–Street–Journal“. Dieses Korpus beinhaltet Artikel dieser Zeitung aus den Jahren 1987–89. Es umfasst ca. 39 Millionen Wörter für das Training der Sprachmodelle und ca. 325 000 Wörter als Testkorpus zur Berechnung der Perplexität. Aus den Trainingstexten wurden weitere kleinere Trainingskorpora von 1 und 4 Millionen Wörtern gebildet. Das Vokabular besteht aus 20 000 Wörtern.

- **NAB:** Das North–American–Business–Korpus ist eine Obermenge von WSJ0 und umfasst ca. 250 Millionen Wörter amerikanischer Zeitungstexte von 1987 bis 1994 für das Training der Sprachmodelle. Dazu kommen ein Development–Korpus (DEV) mit ca. 7000 Wörtern zur Parameteroptimierung und ein Evaluation–Korpus mit ca. 8000 Wörtern als Testkorpus. Auf diesen beiden Korpora lassen sich sowohl Perplexitäten als auch Fehlerraten messen. Das Vokabular besteht ebenfalls aus 20 000 Wörtern.

- **Verbmobil:** Beim Verbmobil–Korpus handelt es sich um deutsche Spontansprache aus dem Bereich der Vereinbarung von Geschäftsterminen. Der Trainingstext umfasst ca. 300 000 Wörter. Weiter gibt es ein Testkorpus mit ca. 7000 Wörtern zur Berechnung der Perplexität und Wortfehlerrate. Das Vokabular umfasst ca. 5000 Wörter.

Kapitel 3

Glättungsverfahren

Die in Kapitel 2.2.1 für Trigramme beschriebenen Glättungsverfahren sollen hier etwas allgemeiner für ein Ereignis (h, w) (z.B. Trigramm: $(h, w) = (u, v, w)$) hergeleitet und ein systematischer experimenteller Vergleich auf großen Texten für Perplexität und Wortfehlerrate durchgeführt werden. Es soll damit untersucht werden, ob eines dieser Verfahren durchgängig die besten Ergebnisse liefert und somit als Standardverfahren für die folgenden Arbeiten verwendet werden kann.

3.1 Parameterschätzung: Maximum–Likelihood und Leaving–One–Out

Wie für Trigramme bereits erwähnt, liefert die Maximum–Likelihood–Schätzung für $p_\theta(w|h)$ (im weiteren Verlauf wird der Übersicht halber nur kurz $p(w|h)$ geschrieben) die relativen Häufigkeiten: Die Normierung $\sum_w p(w|h) = 1$ muß in die ursprüngliche log–Likelihood–Funktion

$$F_{w_1^N}(\theta) = \sum_{h,w} N(h, w) \cdot \log p(w|h)$$

als Nebenbedingung mittels Lagrangescher Multiplikatoren μ_h eingebracht werden:

$$\tilde{F}_{w_1^N}(\theta) = \sum_{h,w} N(h, w) \cdot \log p(w|h) - \sum_h \mu_h \left[\sum_w p(w|h) - 1 \right] \quad .$$

Nullsetzen der Ableitungen

$$\frac{\partial \tilde{F}}{\partial p(w|h)} = \frac{N(h, w)}{p(w|h)} - \mu_h = 0 \quad ,$$

$$\frac{\partial \tilde{F}}{\partial \mu_h} = \sum_w p(w|h) - 1 = 0$$

ergibt als Optimum die relativen Häufigkeiten

$$\mu_h = \sum_w N(h,w) = N(h) \quad ,$$
$$p(w|h) = \frac{N(h,w)}{\mu_h} = \frac{N(h,w)}{N(h)} \quad . \tag{3.1}$$

Für den Fall ungesehener Ereignisse $N(h,w) = 0$ ergibt sich unerwünschterweise $p(w|h) = 0$. Wie bereits in Kapitel 2.2.1 erwähnt, werden Glättungverfahren eingesetzt, um für alle w einer Worthistorie h $p(w|h) > 0$ zu erreichen. Der einfache Fall einer linearen Interpolation

$$p(w|h) = (1-\lambda) \cdot \frac{N(h,w)}{N(h)} + \lambda \cdot \beta(w|\bar{h}) \tag{3.2}$$

läßt sich wie folgt interpretieren: Die relativen Häufigkeiten gehen nur zu einem Anteil $(1-\lambda)$ ein; es bleibt also eine Restmasse λ, die über alle gesehenen und ungesehenen Ereignisse verteilt werden kann. Diese Restmasse wird gemäß einer robusteren, aber auch gröberen Verteilung $\beta(w|\bar{h}) > 0$ auf alle Wörter w aufgeteilt. $\beta(w|\bar{h})$ beruht auf einer verallgemeinerten Worthistorie $\bar{h}$ (z.B. Trigramm→Bigramm: $h = (u,v), \bar{h} = (v)$, $\beta(w|v) = N(v,w)/N(v)$, diese Verteilung kann wiederum durch die Unigramm–Verteilung geglättet werden). Die allgemeinste (und gröbste) Verteilung ist $\beta(w|\bar{h}) = 1/W$ (Zerogramm).

Der Parameter λ wird in diesem Fall Glättungsparameter genannt. Glättungsparameter haben das Problem, daß sie nicht auf dem Trainingskorpus geschätzt werden können, da $N(h,w) \geq 1$ für alle (h,w) des Trainingskorpus gilt und somit nie eine Glättung stattfindet. Die Maximum–Likelihood–Schätzung für λ wäre also $\lambda = 0$. Frühe Arbeiten [Jelinek & Mercer 80, Bahl et al. 83] haben sich durch Kreuzvalidierung [Duda & Hart 73] beholfen: Das Trainingskorpus wird partitioniert, durchlaufen und die Partition, in der man sich gerade befindet („held–out part“), aus dem Trainingskorpus herausgenommen und mit den verbleibenden Partitionen („retained part“) die relativen Häufigkeiten berechnet. Die Leaving–One–Out–Schätzung [Ney et al. 94, Ney et al. 95] ist ein Spezialfall der Kreuzvalidierung. Hier wird nur das aktuelle Ereignis aus der Bestimmung der relativen Häufigkeiten herausgenommen. Gl. (3.2) wird in diesem Fall auf dem Trainingskorpus zu

$$p_{l1o}(w|h) = (1-\lambda) \cdot \frac{N(h,w)-1}{N(h)-1} + \lambda \cdot \beta(w|\bar{h}) \quad .$$

Einmal gesehene Ereignisse, sogenannte Singletons, werden dadurch zu ungesehenen Ereignissen, und Glättung findet statt. Da die meisten Bi– und Trigramm–Ereignisse (h,w) nur einmal im Trainingskorpus vorkommen, ist dies ein sehr effizientes Verfahren, das zudem gut analytisch zu behandeln ist. Abb. 3.1 veranschaulicht das Verfahren.

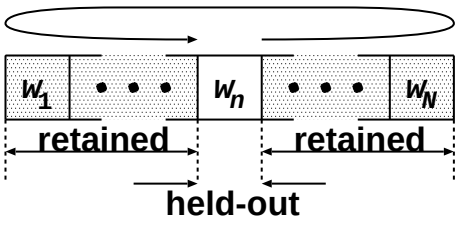

Abbildung 3.1: Schema der Leaving–One–Out–Schätzung.

3.2 Ansätze zur Glättung

3.2.1 Linear Discounting

Beim Linear Discounting wird, wie bereits in Gl. (3.2) vorgestellt, die Wahrscheinlichkeit eines Ereignisses linear verringert, um Wahrscheinlichkeitsmasse für die ungesehenen Ereignisse zu erhalten. Die Verknüpfung der relativen Häufigkeiten mit der Glättungsverteilung $\beta(w|h)$ kann wie in Gl. (3.2) über Interpolation erfolgen, oder aber, motiviert von der Katz–Glättung (siehe Abschnitt 3.2.2), als Auswahl („Backing–Off“):

$$p(w|h) = \begin{cases} (1-\lambda) \cdot \dfrac{N(h,w)}{N(h)}, & N(h,w) > 0, \\ \lambda \cdot \dfrac{\beta(w|\bar{h})}{\sum_{w':N(h,w')=0} \beta(w'|\bar{h})}, & \text{sonst} \end{cases}$$

Damit die Normierungseigenschaft der Wahrscheinlichkeitsverteilung $p(w|h)$ erhalten bleibt, muß hier die Glättungsverteilung $\beta(w|h)$ über die ungesehenen Wörter renormiert werden. Zur Bestimmung des Glättungsparameters für den Backing–Off–Fall lautet die Leaving–One–Out–log–Likelihood–Funktion [Ney et al. 94]:

$$\begin{aligned} F(\lambda) &= \sum_{h,w} N(h,w) \cdot \log[p_{l1o}(w|h)] \\ &= \sum_{h,w:\ N(h,w)>1} N(h,w) \cdot \log\left[(1-\lambda) \cdot \frac{N(h,w)-1}{N(h)-1}\right] \\ &\quad + \sum_{h,w:\ N(h,w)=1} 1 \cdot \log\left[\lambda \cdot \frac{\beta(w|\bar{h})}{\sum_{w':N(h,w')=0} \beta(w'|\bar{h})}\right] \quad . \end{aligned} \tag{3.3}$$

Nullsetzen der Ableitung nach λ ergibt

$$\lambda = \frac{n_1}{N} \tag{3.4}$$

mit $n_1 := \sum_{h,w:\ N(h,w)=1} 1$. Die Interpolation Gl. (3.2) läßt sich nicht geschlossen lösen, Gl. (3.4) ist jedoch eine Näherungslösung dafür.

Gl. (3.3) kann nicht nur zur Bestimmung des Glättungsparameters λ genutzt werden, sondern auch zur analytischen Bestimmung der Glättungsverteilung $\beta(w|h)$ [Kneser & Ney 95]. Die Ableitung ergibt näherungsweise

$$\beta(w|h) = \frac{n_1(\bar{h}, w)}{\sum_{w'} n_1(\bar{h}, w')} \tag{3.5}$$

mit $n_1(\bar{h}, w) = \sum_{h \in \bar{h}:\ N(h,w)=1} 1$. Dies wird als Singleton–Glättungsverteilung bezeichnet.

3.2.2 Katz–Discounting

Das Katz–Discounting [Katz 87] ist eine Weiterentwicklung des Linear Discounting:

1. Es wurde das in Kap. 3.2.1 bereits vorgestellte Backing–Off–Verfahren eingeführt,

2. das Discounting erfolgte nur für Ereignisse mit einer Häufigkeit $N(h, w)$ kleiner einer vorgegebenen Schranke k,

3. jede auf dem Trainingskorpus beobachteten absoluten Häufigkeit $r = N(h, w)$ wird ein eigener Interpolationsparameter λ_r zugeordnet und

4. die Methode zur Schätzung dieses Interpolationsparameters λ_r wurde vorgegeben.

Insgesamt ergibt sich

$$p(w|h) =$$

$$= \begin{cases} \dfrac{N(h,w)}{N(h)} & \text{if } k < N(h,w) \\ \left[1 - \lambda_{N(h,w)}\right] \cdot \dfrac{N(h,w)}{N(h)} & \text{if } 1 \leq N(h,w) \leq k \\ \left(\sum\limits_{w':1 \leq N(h,w') \leq k} \left[\lambda_{N(h,w')} \cdot \dfrac{N(h,w')}{N(h)} \right] \right) \cdot \dfrac{\beta(w|\overline{h})}{\sum\limits_{w':N(h,w')=0} \beta(w'|\overline{h})} & \text{if } N(h,w) = 0 \end{cases}$$

Die Bestimmung der Interpolationsparameter λ_r leitet sich aus der Turing–Good–Schätzung ab, die sich wiederum aus der Leaving–One–Out–Schätzung für Linear Discounting mit Backing–Off für Verbundverteilungen $p(h,w)$ annähern läßt [Ney et al. 95]:

$$\lambda_r = \mu \cdot \left[1 - \frac{(r+1)n_{r+1}}{rn_r}\right]$$

mit $n_r := \sum_{hw:\ N(h,w)=r} 1$ als Anzahl der r–mal gesehenen Ereignisse und μ als Renormalisierungsfaktor. Dieser bestimmt sich aus der geforderten Übereinstimmung der Restmasse mit der Turing–Good–Schätzung für die Wahrscheinlichkeit ungesehener Ereignisse n_1/N

$$\sum_{r=1}^{k} \lambda_r \frac{rn_r}{N} = \frac{n_1}{N}$$

zu

$$\mu = \left[1 - \frac{(k+1)n_{k+1}}{n_1}\right]^{-1} \quad .$$

3.2.3 Absolute Discounting

In [Ney et al. 95] wurde die Beobachtung gemacht, daß für Häufigkeiten $r > 0$ gilt

$$\begin{aligned} \lambda_r \cdot r &\cong b = const(r) \quad \text{, also} \\ (1 - \lambda_r) \cdot r &= r - b \quad . \end{aligned}$$

Durch den häufigkeitsspezifischen Interpolationsparameter λ_r des Katz–Discounting wird also letztendlich die absolute Häufigkeit um einen annähernd festen Betrag, unabhängig von der absoluten Häufigkeit, gekürzt. Die Häufigkeit um einen festen Betrag absolut anstatt linear zu vermindern hat zur Folge, daß die hohen Häufigkeiten, deren zugehörige Ereignisse nach dem Gesetz der großen Zahl auch am besten geschätzt sind, kaum verändert werden und damit sehr nahe an der ursprünglichen Maximum–Likelihood–Schätzung sind. Dies motiviert die Glättungsmethode „Absolute Discoun-

ting“ mit Backing–Off [Ney et al. 94, Ney et al. 95]:

$$p(w|h) = \begin{cases} \dfrac{N(h,w) \; - \; d}{N(h)} & \text{if } N(h,w) > 0 \\ \\ d \cdot \dfrac{W - n_0(h)}{N(h)} \cdot \dfrac{\beta(w|\overline{h})}{\displaystyle\sum_{w':N(h,w')=0} \beta(w'|\overline{h})} & \text{if } N(h,w) = 0 \end{cases}$$

Diese Glättungsmethode entspricht in ihrer Funktionsweise dem Katz–Discounting, kommt aber mit einem Glättungsparameter, dem Discountwert d, aus. Auch dieser Discountwert läßt sich mit Leaving–One–Out aus der log–Likelihood–Funktion schätzen [Ney et al. 94, Ney et al. 95]:

$$F(b) = n_1 \cdot \log \; d + \sum_{r \geq 2} r \cdot n_r \cdot \log \left[r - 1 - d \right] + const(d) \quad .$$

Da der Discountingparameter d additiv in die log–Likelihood–Funktion eingeht, gibt es keine geschlossene Lösung. Angenähert ergibt sich

$$d = \frac{n_1}{n_1 + 2 \cdot n_2} \quad . \tag{3.6}$$

Wie beim Linear Discounting läßt sich auch hier die Singleton–Glättungsverteilung Gl. (3.5) herleiten.

Alternativ zum Backing–Off kann Absolute Discounting auch mit Interpolation verwendet werden [Ney et al. 94]:

$$p(w|h) = \max \left\{ 0, \; \frac{N(h,w) \; - \; d}{N(h)} \right\} \; + \; d \cdot \frac{W - n_0(h)}{N(h)} \cdot \beta(w|\overline{h})$$

mit $n_0(h) := \sum_{w: \; N(h,w)=0} 1$ als Anzahl der Wörter, die auf dem Trainingskorpus nie als Nachfolger der Worthistorie h gesehen worden sind. Eine Lösung der Leaving–One–Out–Schätzung für den Discountingparameter d ist nicht bekannt, es wird daher dieselbe Lösung Gl. (3.6) wie für Backing–Off verwendet.

3.3 Experimentelle Ergebnisse

Die Experimente wurden auf dem Wall–Street–Journal–(WSJ0–)Korpus, dem North–American–Business–(NAB–)Korpus und dem Verbmobil–Korpus durchgeführt. Alle drei Korpora sind im Anhang A beschrieben. Bei den Sprachmodellen handelt es sich um Trigramme, die über Bigramme, Unigramme und, bei ungesehenen Wörtern, über Zerogramm geglättet werden.

In Tab. 3.1 finden sich die Ergebnisse für das WSJ0–Korpus. Mit Zunahme der Länge des Trainingskorpus ist für alle Verfahren ein deutliches Absinken in der Perplexität zu beobachten. Der Unterschied zwischen Interpolation und Backing–Off liegt zwischen 1% bis 4%, meistens ist die Interpolation besser. Größere Bedeutung hat das

Tabelle 3.1: Testperplexitäten für verschiedene Glättungsverfahren, WSJ0.

Glättungsverfahren	1M	4M	39M
Linear Discounting:			
Backing-Off	303.6	201.4	123.1
Interpolation	295.0	196.9	121.4
Katz–Discounting:	250.9	163.5	102.3
Absolute Discounting:			
Backing Off	248.8	163.4	102.5
Interpolation	255.1	168.4	104.9
Absolute Discounting + Singleton–Glättung:			
Backing-Off	238.3	157.5	99.6
Interpolation	229.7	152.1	96.8

eigentliche Glättungsverfahren. Der Unterschied zwischen Linear Discounting und Katz–Discounting liegt zwischen 15% und 17%. Katz–Discounting und Absolute Discounting haben erwartungsgemäß etwa dieselbe Performanz. Eine nochmals deutliche Verbesserung um 6% bis 8% ergibt sich bei Ersetzung der relativen Häufigkeiten bei den Glättungsverteilungen durch die Singleton–Glättungsverteilung. Im Vergleich zum Linear Discounting mit Interpolation als dem ursprünglichen Glättungsverfahren ergibt sich insgesamt eine Reduktion um 20% bis 23%.

Auch auf dem NAB–Korpus in Tab. 3.2 wurden die Trigrammhäufigkeiten und Glättungsparameter auf den 250 Millionen Trainingswörtern bestimmt. Der Skalierungsfaktor (LMScale), mit dem das Sprachmodell im Rescoring gewichtet wird, wurde auf dem DEV–Korpus optimiert. Es zeigt sich die Überlegenheit von Absolute über Linear Discounting durch eine Verminderung von 11–13% in der Perplexität und 4–5% relativ in der Wortfehlerrate auf DEV sowie 14–15% in der Perplexität und 7–9% relativ in der Wortfehlerrate auf EVL. Absolute Discounting gegeben, ist wie bei WSJ0 Backing–Off geringfügig besser (5–6% in der Perplexität und 1–2% in der Wortfehlerrate) als die Interpolation. Durch Hinzufügung der Singleton–Glättungsverteilung gewinnt die Interpolation jedoch so deutlich (8% in der Perplexität und 5% relativ in der Wortfehlerrate auf DEV, 9% in der Perplexität und 1% relativ in der Wortfehlerrate auf EVL) im Vergleich zu Backing–Off (2% in der Perplexität und 2% relativ in der Wortfehlerrate auf DEV, 3% in der Perplexität und -1% relativ in der Wortfehlerrate auf EVL), daß auch hier Absolute Discounting mit Interpolation und Singleton–Glättungsverteilung letztendlich das beste Glättungsverfahren ist.

Die Ergebnisse für Verbmobil in Tab. 3.3 geben dieselben Tendenzen wieder: Der Unterschied zwischen Backing–Off und Interpolation beträgt 8% bis 4% in der Perplexität und 3% bis 5% relativ in der Fehlerrate zugunsten der Interpolation, der Unterschied zwischen Linear und Absolute Discounting 13% in der Testperplexität und 5% relativ in der Fehlerrate. Die Singleton–Glättungsverteilung bringt nur eine Reduktion in der Perplexität um 3%, in der Fehlerrate um 1% relativ. Der Grund ist vermutlich die schlechte

Tabelle 3.2: Testperplexitäten und Wortfehlerraten für verschiedene Glättungsverfahren, NAB.

Glättungsverfahren	LMSc	DEV			EVL		
		PP	Wortfehler [%]		PP	Wortfehler [%]	
			del/ins	WER		del/ins	WER
Linear Discounting:							
Backing–Off	17	144.7	1.7/2.6	13.9	150.4	1.8/3.0	14.8
Interpolation	15	148.8	1.5/3.1	14.1	156.5	1.5/3.4	14.8
Absolute Discounting:							
Backing–Off	17	125.8	1.7/2.4	13.3	127.6	1.9/2.5	13.5
Interpolation	18	132.0	1.5/2.7	13.4	135.4	1.6/2.9	13.8
Absolute Discounting, Singleton–Glättung:							
Backing–Off	19	122.9	1.8/2.0	13.1	123.7	2.1/2.3	13.6
Interpolation	21	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6

Tabelle 3.3: Testperplexitäten und Wortfehlerraten für verschiedene Glättungsverfahren, Verbmobil.

Glättungsverfahren	LMSc	PP	Wortfehler [%]	
			del/ins	WER
Linear Discounting:				
Backing–Off	14	52.4	3.9/3.4	19.6
Interpolation	12	48.2	3.3/4.1	18.9
Absolute Discounting:				
Backing–Off	13	43.8	3.5/3.2	18.8
Interpolation	15	41.9	3.3/3.2	17.9
Absolute Discounting, Singleton–Glättung:				
Backing–Off	13	43.9	3.6/3.0	18.3
Interpolation	14	40.6	3.3/3.1	17.7

Schätzung der Singletons auf diesem sehr kleinem Trainingskorpus. Insgesamt ergibt sich im Vergleich zum Linear Discounting mit Interpolation eine Verbesserung in der Perplexität um 16% und in der Fehlerrate um 6% relativ. Die Fehlerrate fällt dabei mit der Perplexität, einzige Ausnahme ist die Hinzunahme der Singleton–Glättungsverteilung für Backing–Off.

Zusammenfassung. Auf allen Korpora hat sich damit Absolute Discounting mit Interpolation und Singleton–Glättungsverteilung als das beste Verfahren erwiesen. Im Vergleich zum Linear Discounting mit Interpolation als ursprünglichem Glättungsverfahren ergibt sich eine Verbesserung von 16–23% in der Perplexität und 6–9% relativ in der Wortfehlerrate. Für die weiteren Arbeiten wird dieses Verfahren daher als Standard–Trigramm–Sprachmodell gewählt.

Kapitel 4

Varigramme

Wie bereits in Kap. 2.2.2 beschrieben, handelt es sich bei dem Varigramm–Sprachmodell $p(w|h)$ um ein Trigramm–Sprachmodell, das um einzelne m–gramme mit $m > 3$ erweitert ist, d.h. die Historie h hat nunmehr eine unterschiedliche Länge. Das Trigramm–Sprachmodell soll wie in [Ron et al. 94, Niesler & Woodland 96] immer um ganze bedingte Verteilungen, also historienbasiert, anstatt um einzelne Varigramme wie in [Kneser 96] erweitert werden. Es soll untersucht werden, ob die in [Kneser 96] beobachtete Diskrepanz in der Verringerung der Perplexität und Fehlerrate auch mit dieser Auswahlmethode bestehen bleibt und somit der Varigramm–Ansatz zukünftig verwendet werden kann.

4.1 Varigramm–Sprachmodell

Für das Varigramm–Sprachmodell gilt natürlich ebenso die Maximum–Likelihood–Schätzung Gl. (3.1):

$$p(w|h) = \frac{N(h,w)}{N(h)} \quad .$$

Ebenso wie bei den Trigrammen gibt es hier das Sparse–Data–Problem. Dadurch, daß eine Varigramm–Historie eine Spezialisierung ihrer zugrunde liegenden Trigramm–Historie ist und demnach höchstens genauso oft, normalerweise aber noch seltener im Training gesehen werden kann als diese, ist ein gutes Glättungsverfahren sogar noch wichtiger. Wird das ursprüngliche Trigramm–Sprachmodell nicht wie in [Kneser 96] um einzelne Varigramme (h,w), sondern wie in [Ron et al. 94, Niesler & Woodland 96] um ausgewählte Historien h mit allen zugehörigen Varigrammen (h,w) erweitert (d.h. es gilt $\sum_w N(h,w) = N(h)$), so können die in Kap. 3 eingeführten Glättungsverfahren ohne Änderung übernommen werden. Hier wird Absolute Discounting mit Interpolation und Singleton–Glättungsverteilung benutzt.

4.2 Auswahl der Varigramm–Historien

Das Maximum–Likelihood–Kriterium kann nicht nur zur Bestimmung der Parameter eines gegebenen Modells benutzt werden, sondern auch zur Bestimmung des Modells selbst. Soll probehalber die Historie h' um eines ihrer Vorgängerwörter v' auf $h'' := (v', h')$ erweitert werden, so wird das Modell $p(w|h)$ um diese neue Historie h'' probehalber zu $p_{h''}(w|h)$ erweitert. Für die Nachfolgewörter w zu h'' gilt dann nicht mehr die alte Schätzung $p(w|h')$, sondern

$$p_{h''}(w|h'') = \frac{N(h'', w)}{N(h'')} \quad .$$

Für alle anderen (h, w) gilt $p_{h''}(w|h) = p(w|h)$. Damit hat $p_{h''}(w|h)$ auch eine andere log–Likelihood $F_{h''}$, die sich aber nur für (h'', w) von der ursprünglichen log–Likelihood unterscheidet. Die Differenz ist somit

$$\begin{aligned} &\Delta F(h'') := \\ &= F_{h''} - F \\ &= \sum_w N(h'', w) \cdot \log \frac{p_{h''}(w|h'')}{p(w|h')} \\ &= \sum_w N(v', h', w) \cdot \log \left[\frac{N(v', h', w)}{N(v', h')} \cdot \frac{N(h')}{N(h', w)} \right] \quad . \end{aligned} \tag{4.1}$$

Es sollen die erweiterten Historien h'' gewählt werden, die den größten Gewinn in der log–Likelihood erzielen. In der Literatur werden üblicherweise alle existierenden Historien einer Länge um alle ihre beobachteten Vorgängerwörter zu h'' erweitert und die h'' dem Sprachmodell hinzugefügt, deren Gewinn in log–Likelihood $\Delta F(h'')$ über einem fest vorgegebenen Schwellwert κ liegt: $\Delta F(h'') > \kappa$. Mit der nächst höheren Historienlänge wird fortgefahren. Ungeklärt ist aber dabei, wie der Schwellwert κ bestimmt werden soll. Die Vorgehensweise dieser Arbeit ist es daher, unter allen bildbaren Historien $h^{*''}$, unabhängig von der Länge von $h^{*''}$, diejenige Historie h'' herauszufinden, die die log–Likelihood am meisten vergrößert, also

$$h'' = \underset{h^{*''}:\ \Delta F(h^{*''})>0}{\operatorname{argmax}} \left\{ \Delta F(h^{*''}) \right\} \quad .$$

Das vermeidet den Schwellwert κ, ist wegen der Maximumsbildung aber algorithmisch erheblich aufwendiger. Nach der Wahl von h'' werden alle existierenden (h'', w) dem Sprachmodell zugefügt, und die Auswahl beginnt erneut unter Beteiligung aller (v'', h'') mit den Vorgängerwörtern v'' zu h''.

Durch die höhere Historienlänge vergrößert sich auch das bereits erwähnte Sparse–Data–Problem, d.h. von den möglichen Historien ist nur ein winziger Bruchteil in dem Trainingskorpus gesehen worden. Die ausgewählten Historien sind damit zwar optimal auf dem Trainingskorpus, aber nicht unbedingt auf bisher ungesehenen Daten. In [Niesler & Woodland 96] ist ein Leaving–One–Out–Kriterium zum Zweck einer robusteren Schätzung verwendet worden, allerdings wird die genaue Formel nicht angegeben. Für diese Arbeit ist daher die nachstehende Leaving–One–Out–Schätzung für die Auswahl

der Varigramm–Historien auf dem Trainingskorpus entwickelt worden, die auf Absolute Discounting als Glättungsverfahren beruht:

$$p(w|h) = \begin{cases} \dfrac{N(h,w) - d - 1}{N(h) - 1} + (W - n_0(h)) \cdot \dfrac{d}{N(h) - 1} \cdot p(w|\bar{h}) \\ \qquad \text{if } N(h,w) > 1 \text{ and } N(h) > 1 \\ ((W - n_0(h)) - 1) \cdot \dfrac{d}{N(h) - 1} \cdot p(w|\bar{h}) \\ \qquad \text{if } N(h,w) = 1 \text{ and } N(h) > 1 \\ p(w|\bar{h}) \qquad \text{if } N(h) = 1 \end{cases}$$

Wie in Kapitel 3 bezeichnet $\bar{h}$ die um ein Wort verkürzte Historie h, d.h. es gibt ein Wort v aus dem Vokabular mit $h = (v, \bar{h})$. Diese Schätzung ersetzt in Gl. (4.1) die relativen Häufigkeiten. Zu beachten ist, daß Singleton–Historien vollständig aus der Auswahl herausfallen.

4.3 Experimentelle Ergebnisse

Tabelle 4.1 gibt eine Übersicht über die jeweils besten Testperplexitäten für Varigramme mit und ohne Leaving–One–Out–Auswahl im Vergleich zum Trigramm–Modell als Ausgangspunkt auf WSJ0. Die Varigramme zeigen eine deutliche Perplexitätsverminderung für das große 39M–Trainingskorpus um 9%, jedoch kaum für das kleinere Teilkorpus. Auf WSJ0–1M haben sich überhaupt keine Verbesserungen ergeben, weshalb diese Ergebnisse weggelassen wurden. Offenbar haben die kleineren Teilkorpora einen zu geringen Umfang für eine statistisch signifikante Historienauswahl. Ferner gibt es kaum Unterschiede in der Testperplexität zwischen der reinen Maximum–Likelihood–Auswahl und der Leaving–One–Out–Auswahl. Diese Testperplexität wird jedoch für die Leaving–One–Out–Auswahl mit nur einem Drittel der gewählten Historien und nur einem Viertel der daraus resultierenden Varigramme erreicht. Tabelle 4.2 zeigt die zehn zuerst gewählten Historien und die fünf zuerst gewählten Historien mit einer Länge größer drei. Es lassen sich drei Charakteristika feststellen:

1. Es handelt sich um häufige Historien. Dies erklärt auch den geringen Unterschied zwischen Leaving–One–Out und reiner Maximum–Likelihood, da häufige Historien h'' in der Regel auch häufige Varigramme (h'', w) beinhalten und Leaving–One–Out darauf dann kaum Auswirkung hat.

2. Es handelt sich um Historien mit einem Zwei–Wort–Begriff („NEW YORK“, „U. S.“), für die der Varigramm–Ansatz auch motiviert wurde.

3. Es handelt sich um Historien, die in einem sehr allgemeinen Wort enden („THE“), nach welchem sehr viele Nachfolgerwörter möglich sind. Diese Anzahl Nachfolgerwörter kann durch eine höhere Historienlänge eingeschränkt werden.

Tabelle 4.1: Anzahl ausgewählter Historien H und Varigramme, WSJ0–4M und WSJ0–39M, mit und ohne L1O–Auswahl sowie Testperplexitäten.

L1O	H		Varigramme		PP_{Test}	
	4M	39M	4M	39M	4M	39M
—	0	0	0	0	152.1	96.8
nein	10 000	900 000	255 416	9 101 930	150.1	89.0
ja	10 000	300 000	80 892	2 148 328	148.4	88.2

Tabelle 4.2: L1O–gewählte Historien, WSJ0–39M.

Pos.	$N(h'')$	$\Delta F(h'')$	Text
1	9676	12508	ONE OF THE
2	41640	11144	THE U. S.
3	11609	8906	PERCENT OF THE
4	14164	5817	</s> IN THE
5	2542	5323	TRADING ON THE
6	14675	5099	</s> THE COMPANY
7	2937	5081	END OF THE
8	3015	4595	DOLLAR A SHARE
9	5706	4485	</s> FOR THE
10	4413	4411	ACCORDING TO THE
65	3283	1993	</s> IN NEW YORK
78	3491	1839	ON THE NEW YORK
87	10036	1746	IN THE U. S.
146	596	1210	EXCHANGE IN NEW YORK
266	1801	827	THAT THE U. S.

Von den gewählten Historien haben die meisten die Länge drei (Tabelle 4.3), danach nimmt mit zunehmender Länge diese Zahl um je eine Größenordnung ab. Den größten Einfluß haben also die Erweiterungen von Trigrammen zu Viergrammen. Aufgrund der effizienteren Parameterschätzung ist für die weiteren Arbeiten nur noch das Leaving–One–Out–Kriterium verwendet worden.

Die Messung der Wortfehlerrate wurde auf dem NAB–Korpus durchgeführt. Dabei mußten zwei technisch bedingte Einschränkungen hingenommen werden:

- Die Auswahl der besten Varigramm–Historie impliziert, daß auf das gesamte Korpus zugegriffen werden muß. Bei den 250 Millionen Wörtern des NAB–Korpus war dies aus Speicherplatzgründen nicht möglich, so daß mit einem Auszug von ca. 40 Millionen Wörtern, dessen Artikel gleichmässig über das NAB–Korpus verteilt sind, gearbeitet werden mußte. Da aus Tabelle 4.2 hervorgeht, daß die gewählten

Tabelle 4.3: Verteilung der Historienlängen für L1O–bestimmte Varigramme, WSJ0.

Historienlänge	4M	39M
3	9 351	246 513
4	601	47 231
5	44	5 742
6	4	493
7	–	19
8	–	2
Summe	10 000	300 000

Historien auch häufige Historien sind, kann man davon ausgehen, daß sie auch in diesem Auszug in ausreichender Häufigkeit vertreten sind.

- Das Rescoring auf dem Wortgraphen muß nun ebenfalls die längere Historie berücksichtigen, was zu einer erheblichen Verlängerung der Laufzeiten führen kann. Da aus Tabelle 4.3 klar hervorgeht, daß etwa 80% der gewählten Varigramm–Historien die Länge drei haben, wurden für das NAB–Korpus auch nur solche Historien ausgewählt und ein Viergramm–Rescoring implementiert.

Die ausgewählten Historien ähneln den auf WSJ0 gewählten und werden deshalb nicht noch einmal angegeben.

Tabelle 4.4 zeigt Perplexität und Wortfehlerrate für verschiedene Anzahlen ausgewählter Varigramm–Historien auf den DEV–Daten. Während die Perplexitäten nicht so stark wie auf WSJ0, aber doch deutlich fallen, erreicht die Wortfehlerrate bei 10 000 gewählten Historien ein kaum merkliches Minimum, obwohl die Perplexitäten weiter fallen. Die Wiederholung des Experiments auf den EVL–Daten mit den auf DEV optimierten Parametern (Skalierungsfaktor und Historienanzahl) in Tabelle 4.5 erbrachte erwartungsgemäß auch nur eine geringe Verbesserung. Warum die Verbesserung der Wortfehlerrate geringer ausfällt als die Verbesserung der Perplexität, konnte im Rahmen dieser Arbeit nicht genau geklärt werden. Eine Vermutung ist, daß die ausgewählten Varigramm–Historien nur sehr wenige nachfolgende Wörter mit einer entsprechend hohen Wahrscheinlichkeit haben, so daß die entsprechenden Wortfolgen trotz geringerer akustischer Evidenz im Rescoring eine hohe Bewertung erhalten und die mit dem Trigramm–Modell noch korrekt erkannten Wortfolgen verdrängen. Darauf weist auch der im Vergleich zu den Trigrammen etwas geringere optimale Skalierungsfaktor für Varigramme hin, der ein Hinweis auf die Qualität des Sprachmodells ist.

Für Verbmobil finden sich die besten Ergebnisse, sowohl für Testperplexität als auch für Fehlerrate, in Tabelle 4.6. Im Gegensatz zum NAB–Korpus fallen das Optimum für Perplexität und Fehlerrate zusammen. Insgesamt ergibt sich jedoch gegenüber dem Trigramm kaum eine Verbesserung, sowohl in der Perplexität, wie es auch schon auf den kleineren WSJ0–Korpora beobachtet worden war, als auch in der Fehlerrate, ähnlich den Ergebnissen auf NAB. Im Gegensatz zu NAB war es auf dem kleinen Verbmobil–Korpus

Tabelle 4.4: Testperplexitäten und Wortfehlerraten für Varigramm–Modelle, NAB–DEV.

Modell	H	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Trigramm	—	21	121.8	1.8/2.0	12.8
Varigramm	2 000	19	119.4	1.6/2.1	12.7
	5 000	19	118.5	1.6/2.1	12.7
	10 000	19	117.9	1.6/2.1	12.7
	20 000	19	117.1	1.7/2.1	12.8

Tabelle 4.5: Testperplexitäten und Wortfehlerraten für Varigramm–Modelle, NAB–EVL, 10 000 gewählte Historien, LMSc 19.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Trigramm	123.4	2.0/2.3	13.6
Varigramm	119.2	2.0/2.3	13.4

möglich, ein vollständiges Viergramm–Sprachmodell zu erzeugen, indem jede beobachtete Historie des ursprünglichen Trigramm–Modells um ihre Vorgängerwörter erweitert wurde. Sowohl in Perplexität als auch in Wortfehlerrate hat das Viergramm etwas bessere Ergebnisse als das Varigramm, was darauf hindeutet, daß die Auswahl noch nicht optimal ist, was entweder an dem Kriterium oder aber auch an dem kleinen Korpus liegen kann. Insgesamt ist aber auch das Viergramm nicht viel besser als das ursprüngliche Trigramm, daher ist vermutlich auch der Varigramm–Ansatz auf Verbmobil nicht mehr deutlich zu verbessern. Die gewählten Historien für Verbmobil in Tabelle 4.7 lassen sich nicht so genau charakterisieren wie diejenigen der beiden englischsprachigen Korpora. Tendenziell stehen Wörter am Historienende, auf die sehr viele Wörter folgen können („der“, „es“, „denn“), so daß durch die Verlängerung der Historie eine Einschränkung der Auswahl vorgenommen wird.

Tabelle 4.6: Testperplexitäten und Wortfehlerraten für Trigramm, Varigramm (maximale Historienlänge drei) und Viergramm, Verbmobil (H: Anzahl Historien).

Modell	H	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Trigramm	—	14	40.6	3.3/3.1	17.7
Varigramm	500	18	39.5	3.7/2.7	17.5
Viergramm	—	17	39.3	3.6/2.7	17.3

Tabelle 4.7: L1O–gewählte Historien, Verbmobil.

Pos.	$N(h'')$	$\Delta F(h'')$	Text
1	117	155.42	Arbeitstreffen in der
2	637	127.75	</s> <NIB> <NIB>
3	584	98.67	wie wäre es
4	78	84.20	also bei mir
5	129	67.42	Ihnen einen Termin
6	34	66.88	wann wäre es
7	158	65.32	sieht es denn
8	186	61.42	da bei Ihnen
9	39	54.82	halten Sie denn
10	64	54.38	mir wäre es

Zusammenfassung. Zusammenfassend läßt sich sagen:

- Durch den Varigramm–Ansatz kann die Testperplexität um bis zu 9% gesenkt werden, vor allem auf großen Korpora.
- Durch den Einsatz des Leaving–One–Out–Kriteriums läßt sich die Zahl der zum ursprünglichen Trigramm hinzugenommenen Varigramme vierteln.
- Die Reduktion in der Perplexität schlägt sich nicht in einer angemessenen Reduktion der Fehlerrate nieder.

Da das Varigramm zusätzlich eine Viergramm–Suche benötigt (oder für größere Längen eine Suche entsprechend höherer Ordnung), handelt es sich um ein aufwendiges Verfahren. Dieser Aufwand wird durch die geringen Verbesserungen kaum gerechtfertigt, so daß der Varigramm–Ansatz keine geeignete Ergänzung des Trigramm–Modells ist.

Kapitel 5

Wortphrasen

Das Prinzip der Wortphrasen ist bereits in Kapitel 2.2.3 kurz vorgestellt worden: Geeignete Wortpaare werden zu einer Wortphrase verbunden, die als neues Wort in das Vokabular eines Trigramm–Sprachmodells aufgenommen wird. Dadurch wird eine Kontexterweiterung erzielt, ähnlich wie bei den Varigrammen. Bei Varigrammen bleibt das Vokabular gleich und der Kontext wird explizit erweitert. Bei den Wortphrasen bleibt die Kontextlänge gleich, aber das Vokabular wird ergänzt. Es soll auf großen Korpora der Effekt der Wortphrasen untersucht werden, insbesondere, inwieweit das Auswahlverfahren Einfluß auf die Perplexität hat und ob das in Kapitel 2.2.3 erwähnte Summenkriterium zur Bestimmung der Satzwahrscheinlichkeit aus Wortphrasen nicht durch einfachere Verfahren ersetzt werden kann. Auch soll untersucht werden, ob, anders als die Varigramme, die Wortphrasen auch einen positiven Effekt auf die Fehlerrate haben und somit die bessere Alternative für die Kontexterweiterung sind.

5.1 Phrasen–Sprachmodell

Je nachdem, ob zur Auswahl der Phrasen nur stets das ursprüngliche Vokabular herangezogen wird oder aber das bereits um Wortphrasen erweiterte, wird in dieser Arbeit von „flachen“ oder „hierarchischen“ Wortphrasen gesprochen. Im Fall der hierarchischen Auswahl können auch Wortphrasen mit einer Länge größer zwei entstehen, wie in diesem Beispiel:

$$\begin{aligned} \text{AS GOOD} &\rightarrow \text{AS\_GOOD} \\ \text{AS\_GOOD AS} &\rightarrow \text{(AS\_GOOD)\_AS} \end{aligned}$$

In dieser Arbeit verbindet der Unterstrich zwei (oder mehr) Wörter zu einer Phrase. Die Klammerung zeigt, in welcher Reihenfolge die Phrase bei der hierarchischen Auswahl zusammengesetzt wurde. Für das resultierende Sprachmodell ist diese Reihenfolge jedoch unerheblich, und die Klammern werden dort weggelassen.

Da in aller Regel die an einer Phrase beteiligten Wörter oder Teilphrasen auch in anderen Kontexten vorkommen, können sie nach erfolgter Phrasenbildung nicht aus dem

Vokabular entfernt werden. Damit ist die Abbildung von Phrasen auf die Wörter eines gegebenen Textes nicht eindeutig. Um bei dem obigen Beispiel zu bleiben: Steht die Wortfolge „... AS GOOD AS ...“ in einem Text so gibt es drei Möglichkeiten der Abbildung: „AS GOOD AS“ (Wörter sind Phrasen der Länge eins), „AS_GOOD AS“ und „AS_GOOD_AS“. Da alle drei Abbildungen zulässig sind, müssen sie auch alle berücksichtigt werden. Die Abbildung „AS GOOD_AS“ ist nicht bildbar, da im obigen Beispiel keine Wortphrase „GOOD_AS“ gewählt wurde.

Einer Folge von Wortphrasen läßt sich eindeutig eine Wortfolge zuordnen. Umgekehrt kann, wie oben gesehen, eine Wortfolge durch mehrere Phrasenfolgen dargestellt werden. Damit läßt sich die Menge aller bildbaren Phrasenfolgen eindeutig in Klassen partitionieren, so daß jede Klasse einer bildbaren Wortfolge entspricht und ihre Elemente diejenigen Phrasenfolgen sind, die sich der Wortfolge eindeutig zuordnen lassen. Aufgrund der Konstruktion der Wortphrasen hat jede Klasse nur endlich viele Elemente. Wird jedem Element der Klasse nun eine Wahrscheinlichkeit zugeordnet, so ist die Wahrscheinlichkeit der Klasse die Summe der Wahrscheinlichkeiten ihrer Elemente, die Wahrscheinlichkeit einer Wortfolge also die Summe der Wahrscheinlichkeiten aller ihr zugeordneten Folgen von Wortphrasen. Dies geschieht in Analogie zu den stochastischen Grammatiken, wo ein Satz mehrere syntaktische Ableitungen („Parses“) haben kann und die Satzwahrscheinlichkeit der Summe der Wahrscheinlichkeiten dieser Ableitungen entspricht [Jelinek et al. 90]. Diese Vorgehensweise wird „Summenkriterium“ genannt, in Abgrenzung zu später noch zu erläuternden Approximationen.

Somit muß jede Wortfolge auf alle bildbaren Arten unterteilt werden und die Wahrscheinlichkeiten derjenigen Unterteilungen, die einer Folge von Wortphrasen entsprechen, aufsummiert werden. Das Problem, alle bildbaren Unterteilungen zu betrachten, ist in dieser Arbeit effizient mit Hilfe der dynamischen Programmierung gelöst worden, ähnlich der Vorgehensweise in der akustischen Suche [Ortmanns et al. 97]. Eine Wortfolge w_1^N wird also in i ($1 \leq i \leq N$) Teilfolgen der Längen l_j ($1 \leq j \leq i$) unterteilt, wobei $j = 1$ die erste Teilfolge bezeichnet. Das erste Wort der Teilfolge j hat den Index $s_j = \sum_{j'=1}^{j-1} l_{j'}$. Bilden die Wörter $w_{s_j}^{s_j+l_j-1}$ der Teilfolge j eine Wortphrase, so wird dies mit $[w_{s_j}^{s_j+l_j-1}]$ oder auch abkürzend mit π_j gekennzeichnet. Damit gilt für ein wortphrasenbasiertes Trigramm–Sprachmodell:

$$
\begin{aligned}
p(w_1^N) &= \sum_{i=1}^{N} \sum_{\substack{(l_1,\ldots,l_i) \\ \pi_j \text{ existiert, } 1\leq j\leq i}} p(\pi_1^i) \\
&= \sum_{i=1}^{N} \sum_{\substack{(l_1,\ldots,l_i) \\ \pi_j \text{ existiert, } 1\leq j\leq i}} \prod_{j=1}^{i} p(\pi_j|\pi_1^{j-1}) \\
&= \sum_{i=1}^{N} \sum_{\substack{(l_1,\ldots,l_i) \\ \pi_j \text{ existiert, } 1\leq j\leq i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \quad .
\end{aligned}
$$

Zur effizienten Berechnung der Wahrscheinlichkeit für die Wortfolge $p(w_1^N)$ werden die Größen $Q(n)$ und $Q(n,l,l')$ eingeführt. $Q(n)$ ist die Wahrscheinlichkeit für die ersten n Wörter der Wortfolge w_1^N, $Q(n,l,l')$ steht für die Wahrscheinlichkeit der ersten n Wörter, wovon die letzten l Wörter und die l' Wörter vor diesen jeweils eine Phrase bilden. Es gilt also:

$$\begin{aligned}
Q(n) &= p(w_1^n) \\
&= \sum_{i=1}^{n} \sum_{\substack{(l_1,\ldots,l_i) \\ \pi_j \text{ existiert, } 1\le j\le i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{l,l'} Q(n,l,l') \\
Q(n,l,l') &= \sum_{i=1}^{n} \sum_{\substack{(l_1,\ldots,l_i) \\ l_i=l,\ l_{i-1}=l' \\ \pi_j \text{ existiert, } 1\le j\le i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{l''=1}^{n-l-l'} p([w_{n-l+1}^{n}]|[w_{n-l-l'-l''+1}^{n-l-l'}],[w_{n-l-l'+1}^{n-l}]) \cdot Q(n-l,l',l'') \qquad (5.1)
\end{aligned}$$

mit der Konvention $p([w_{n-l+1}^{n}]|[w_{n-l-l'-l''+1}^{n-l-l'}],[w_{n-l-l'+1}^{n-l}]) = 0$ falls $[w_{n-l-l'-l''+1}^{n-l-l'}]$, $[w_{n-l-l'+1}^{n-l}]$ oder $[w_{n-l+1}^{n}]$ nicht existieren. Damit gilt $Q(n,l,l') = 0$ für die meisten n,l,l', jedoch gilt stets $Q(n,1,1) > 0$, wegen $([w_{n-1}^{n-1}],[w_n^n]) = (w_{n-1},w_n)$ stets existent. Somit ist $Q(n) > 0$ und damit auch $p(w_1^N) = Q(N) > 0$. Die Herleitung von Gleichung (5.1) ist in Anhang C beschrieben. Durch das Speichern der Zwischenergebnisse in $Q(n,l,l')$ lassen sich die 2^{N-1} möglichen Aufteilungen der Wortfolge w_1^N in grob

$$\sum_{n=1}^{N} \sum_{l=1}^{n-1} \sum_{l'=1}^{n-l} \sum_{l''=1}^{n-l-l'} 1 \approx \frac{N^4}{24}$$

Schritten berechnen. Durch eine geschickte Implementierung brauchen nur die $Q(n,l,l') > 0$ betrachtet werden. Im günstigsten Fall, in dem nur Phrasen der Länge eins vorkommen, läßt sich so das Summenkriterium linear in N berechnen. Da in der Praxis längere Phrasen selten vorkommen, läßt sich somit das Summenkriterium auch im Durchschnittsfall mit annähernd linearem Aufwand berechnen.

Mit diesem Verfahren ist die Gesamtwahrscheinlichkeit $p(w_1^N)$ berechnet worden. In der Spracherkennung wird aber die bedingte Wahrscheinlichkeit $p(w_n|h_n)$ benötigt. Motiviert von [Jelinek et al. 90] ist daher für diese Arbeit eine Left–to–Right–Inside–Gleichung für Wortphrasen entwickelt worden, die in Anhang D beschrieben ist. Wie später noch erläutert wird, ist diese Gleichung jedoch nicht in den Experimenten verwendet worden.

Aus Gründen der numerischen Stabilität wird in der Praxis der Logarithmus von $Q(n,l,l')$ verwendet. Da sich der Logarithmus einer Summe nicht aufspalten läßt, verkompliziert dies die Implementierung von Gl. (5.1), da die $Q(n-l,l',l'')$ jedesmal zurückgerechnet werden müssen. Die Summenbildung kann durch eine Vereinfachung umgangen werden, der sogenannten „Maximum–Approximation“. Hier wird die Summation durch

eine Maximumsbildung ersetzt, d.h. die Summe der Wahrscheinlichkeiten aller bildbaren Phrasenfolgen wird durch die größte Einzelwahrscheinlichkeit unter den bildbaren Phrasenfolgen angenähert:

$$\begin{aligned} p(w_1^N) &\approx \max_{i=1}^{N} \max_{\substack{(l_1,\ldots,l_i) \\ \pi_j \text{ existiert, } 1\le j\le i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\ &= Q(N) \\ Q(n) &= \max_{l,l'} Q(n,l,l') \\ Q(n,l,l') &= \max_{l''=1}^{n-l-l'} p([w_{n-l+1}^{n}]|[w_{n-l-l'-l''+1}^{n-l-l'}],[w_{n-l-l'+1}^{n-l}]) \cdot Q(n-l,l',l''). \end{aligned} \tag{5.2}$$

Wegen der Monotonie des Logarithmus gilt Gl. (5.2) auch für $\log(Q(n,l,l'))$. Die Anzahl der Berechnungsschritte verändert sich nicht.

In einer weiteren Vereinfachung, im Folgenden „Maximale Abdeckung“ genannt, wird nicht die wahrscheinlichste, sondern die kürzeste Phrasenkette gewählt, d.h. diejenige Phrasenkette, bei der die Wörter maximal durch Phrasen abgedeckt werden:

$$\begin{aligned} \tilde{i} &:= \min_i \{i : (l_1 ... l_i), \pi_j \text{ existiert für } 1 \le j \le i\} \\ p(w_1^n) &\approx \prod_{j=1}^{\tilde{i}} p(\pi_j|\pi_{j-2},\pi_{j-1}) \quad . \end{aligned}$$

Gibt es mehrere Phrasenketten mit minimaler Länge, so wird unter diesen eine zufällig ausgewählt. Die Methode der Maximalen Abdeckung hat den Vorteil, daß zur Bestimmung dieser Kette gar keine Wahrscheinlichkeiten benötigt werden und daher dieses Kriterium sehr einfach zu implementieren ist[1]. Auch die kürzeste Phrasenkette läßt sich mit dynamischer Programmierung bestimmen. Dazu bezeichne $\tilde{i}(n)$ die minimale Anzahl Phrasen, die die Wortkette bis zur Position n einschließlich abdeckt:

$$\begin{aligned} \tilde{i}(0) &:= 0 \\ \tilde{i}(n) &= \min_{j=0\ldots n-1:\ [w_{n-j}^n] \text{ existiert}} \tilde{i}(n-j-1)+1 \quad . \end{aligned}$$

D.h. über alle Phrasen, die in der Position n enden, wird die minimale Anzahl an Vorgängerphrasen gesucht.

Durch die Wortphrasenbildung vergrößert sich das Vokabular und verkleinert sich das Korpus auf die Länge $\tilde{N}$. Die Perplexität der Wortfolge w_1^N bezieht sich natürlich weiterhin auf das ursprüngliche Vokabular und die ursprüngliche Korpuslänge N und ist nicht zu verwechseln mit der phrasenbasierten Perplexität

$$\tilde{PP} = \left[p(\pi_1^{\tilde{N}})\right]^{-\frac{1}{\tilde{N}}} \quad ,$$

die sich bei einer eindeutigen Abbildung der Wortfolge w_1^N auf die Phrasenfolge $\pi_1^{\tilde{N}}$ errechnen läßt, z.B. bei der Maximum–Approximation oder der maximalen Abdeckung.

[1] Dies ist bei der Korpusaufbereitung von sehr großem Vorteil.

Wegen der Setzung $p(\pi_1^{\tilde{N}}) = p(w_1^N)$ läßt sich jedoch die wortbasierte Perplexität durch die Renormierung

$$PP = \tilde{P}P^{\frac{\tilde{N}}{N}}$$

aus der phrasenbasierten Perplexität leicht berechnen.

5.2 Auswahl der Wortphrasen

Wie die Varigramme werden auch die Wortphrasen durch Maximum–Likelihood–Schätzung ausgewählt. Als Ansatz sei die Verbesserung in der Unigramm–log–Likelihood

$$\begin{aligned} F &= \sum_w N(w) \cdot \log p(w) \\ &= \sum_w N(w) \cdot \log \left[\frac{N(w)}{N} \right] \end{aligned}$$

betrachtet. Ist F die ursprüngliche log–Likelihood und $N(\cdot)$ eine ursprüngliche Häufigkeit, so bezeichne $\tilde{F}(a,b)$ die neue Unigramm–log–Likelihood und $\tilde{N}(\cdot)$ die neue Häufigkeit, falls das Vokabular um die Wortphrase $c = (a,b)$ mit a und b als Wörter aus dem Ausgangsvokabular erweitert wurde. Es gilt dann:

$$\begin{aligned} \tilde{N}(w) &= N(w), \quad w \neq a,b,c \\ \tilde{N}(a) &= N(a) - N(a,b) \\ \tilde{N}(b) &= N(b) - N(a,b) \\ \tilde{N}(c) &= N(a,b) \\ \tilde{N} &= N - N(a,b) \end{aligned}$$

Somit läßt sich die Differenz in der log–Likelihood effizient berechnen:

$$\begin{aligned} & \Delta F(a,b) \\ &= \tilde{F}(a,b) - F \\ &= \sum_{w \neq a,b,c} \tilde{N}(w) \cdot \log \tilde{p}(w) + \tilde{N}(a) \cdot \log \tilde{p}(a) + \tilde{N}(b) \cdot \log \tilde{p}(b) + \tilde{N}(c) \cdot \log \tilde{p}(c) \\ &\quad - \sum_w N(w) \cdot \log p(w) \\ &= [N(a) - N(a,b)] \cdot \log [N(a) - N(a,b)] + [N(b) - N(a,b)] \cdot \log [N(b) - N(a,b)] \\ &\quad + N(a,b) \cdot \log N(a,b) - [N - N(a,b)] \cdot \log [N - N(a,b)] \\ &\quad - N(a) \cdot \log N(a) - N(b) \cdot \log N(b) + N \cdot \log N \end{aligned} \tag{5.3}$$

Dieses Kriterium gilt strenggenommen nur für Wörter $a \neq b$. Da es in der Praxis aber recht selten ist, daß zwei gleiche Wörter Kandidaten als Wortphrase sind, wird diese Ungenauigkeit in dieser Arbeit akzeptiert.

In dieser Arbeit werden zwei Vorgehensweisen betrachtet, „flache“ und „hierarchische“ Auswahl genannt. Bei der flachen Auswahl wird das log–Likelihood–Kriterium auf alle

Wortpaare des ursprünglichen Vokabulars angewandt und die k besten Paare zur Vokabularerweiterung verwendet. Bei der hierarchischen Auswahl werden die bisherigen Wortphrasen stets mitberücksichtigt. Dadurch kann sich durch die geänderten und im Kriterium verwendeten Häufigkeiten $\tilde{N}(\cdot)$ die Reihenfolge der weiteren Auswahl ändern. Weiter sind hier auch Wortphrasen, die aus mehr als drei Wörtern bestehen, möglich. Durch die Anpassung der Häufigkeiten $\tilde{N}(\cdot)$, vor allem der Korpuslänge $\tilde{N}$, ändert sich die Bewertung für alle Wortpaare, und $\Delta F(a, b)$ muß nach jeder Erweiterung für alle existierenden Wortpaare (a, b) neu berechnet werden. Dadurch ist die Implementierung aufwendiger und die Laufzeit deutlich länger als bei der flachen Auswahl. Da sich in den Experimenten zeigen wird, daß stets häufige Wortpaare ausgewählt werden, kann die Laufzeit durch hohe Schwellwerte für die betrachteten Wortpaare reduziert werden, ohne daß sich die Ergebnisse dadurch verschlechtern.

Weil in der Sprachmodellierung in dieser Arbeit grundsätzlich Trigramme verwendet werden, sollte das Auswahlkriterium ebenfalls trigramm– statt unigrammbasiert sein. Jedoch ist schon bereits das Bigrammkriterium, das für diese Arbeit hergeleitet wurde und in Anhang E beschrieben ist, sehr aufwendig. Wie später gezeigt wird, steht dieser Aufwand in keinem vernünftigen Verhältnis zu den dadurch erzielten Verbesserungen. Deshalb wurde auf die Entwicklung eines Trigramm–Kriteriums verzichtet. Desweiteren wurde für das Unigramm–log–Likelihood–Kriterium auch eine Leaving–One–Out–Variante entwickelt. Diese ist in Anhang F beschrieben.

5.3 Experimentelle Ergebnisse

Tabelle 5.1 gibt die mit dem hierarchischen Unigramm–log–Likelihood–Kriterium Gl. (5.3) gewählten Wortpaare an. Als Nebeneffekt der hierarchischen Auswahl ändert sich, wie bereits in Kap. 5.2 erwähnt, durch eine Wortphrasenbildung und die damit verbundene Anpassung der Häufigkeiten die Bewertung aller übrigen Wortpaare. Deshalb fällt der Gewinn in log–Likelihood in Tabelle 5.1 nicht kontinuierlich ab. Es lassen sich Parallelen zu den gewählten Varigramm–Historien in Tabelle 4.2 ziehen:

- Es handelt sich stets um sehr häufige Wortpaare.
- Die Wortpaare sind entweder zusammenhängende Begriffe wie z.B. „U. S.“, „NEW YORK“ und „NEW YORK STOCK EXCHANGE“ oder
- die Wortpaare enden in einem sehr allgemeinen Wort („THE“, „THAN“), nach dem sehr viele Wörter möglich sind. Durch die Wortphrase ist effektiv ein Viergramm entstanden, das die Menge dieser Nachfolgerwörter einschränkt.

Für die Untersuchung der geeigneten Berechnung der Wahrscheinlichkeiten nach dem Summenkriterium, der Maximum–Approximation oder der Maximalen Abdeckung wurden die nach dem flachen Unigram–log–Likelihood–Kriterium auf WSJ0–39M gewählten Wortphrasen nach dem Prinzip der Maximalen Abdeckung in den Text des Trainingskorpus eingebracht und daraus ein gewöhnliches Trigramm–Sprachmodell auf Grundlage

Tabelle 5.1: Beispiele für gewählte Wortpaare, Gewinn in log–Likelihood und Wortpaarhäufigkeiten für hierarchisches Unigramm–log–Likelihood–Kriterium, WSJ0–39M.

No.	$\Delta F(a,b)$	$N(a,b)$	$a \quad b$
1	513996	88350	U. S.
2	473020	109275	MILLION DOLLARS
3	444182	83011	NINETEEN EIGHTY
4	233704	39201	NEW YORK
5	215836	42897	BILLION DOLLARS
6	191602	63269	ONE HUNDRED
7	143677	71098	MR. <UNK>
8	130582	28356	MORE THAN
9	128992	215526	OF THE
10	138047	172298	IN THE
12	134986	23929	CENTS A_SHARE
28	65539	8911	CHIEF_EXECUTIVE OFFICER
35	62726	26987	IN NINETEEN_EIGHTY
49	54429	9119	NEW_YORK STOCK_EXCHANGE
54	51877	9045	DOLLARS A_SHARE

des um die Wortphrasen erweiterten Vokabulars erstellt. Mit diesem Sprachmodell wurde die Perplexität des Testkorpus auf die drei möglichen Arten berechnet, das Ergebnis findet sich, zurückgerechnet von den Wortphrasen auf die Wörter, in Tabelle 5.2. Wie zu erwarten ist das Summenkriterium die beste und die Maximale Abdeckung die schlechteste Methode. Jedoch sind die Unterschiede so gering, daß sich der Aufwand für die beiden komplexeren Berechnungen nicht lohnt. Somit ist für den Rest dieser Arbeit nur das Prinzip der Maximalen Abdeckung verwendet worden. Insbesondere wurde auch auf ein Training der Wahrscheinlichkeiten nach dem Summenkriterium oder der Maximum–Approximation verzichtet, da keine wesentlichen Verbesserungen zu erwarten waren. Dies befindet sich in Übereinstimmung mit der Literatur [Deligne & Bimbot 95].

Die Auswirkung der Korpusgröße gibt Tabelle 5.3 für WSJ0-1M und WSJ0-4M in Zusammenhang mit Tabelle 5.2 für WSJ0-39M. Die Wortphrasen wurden jeweils auf den Teilkorpora ausgewählt, mit sehr ähnlichen resultierenden Wortphrasen. Aus den Testperplexitäten ist zu ersehen, daß ein deutlicher Gewinn mit 7% nur auf dem größten Teilkorpus existiert. Die kleineren Teilkorpora verbessern sich hingegen kaum, verschlechtern sich aber bereits bei einer noch recht geringen Anzahl gewählter Phrasen. Offenbar findet durch die Hinzunahme der Phrasen auf den ohnehin stark untertrainierten Korpora eine weitere Überspezialisierung statt mit negativen Konsequenzen für die Testperplexität.

In Tabelle 5.4 findet sich ein Vergleich der in Kap. 5.2 beschriebenen verschiedenen Auswahlkriterien für Wortphrasen. Die Testperplexitäten unterscheiden sich nur geringfügig,

Tabelle 5.2: Testperplexität, WSJ0–39M, flaches Unigramm–Kriterium, für unterschiedliche Phrasenanzahlen P und die drei möglichen Arten der Berechnung.

P	Summen–Kriterium	Maximum–Approximation	Maximale Abdeckung
0	96.8		
500	91.2	91.2	91.2
1000	90.5	90.5	90.5
2000	90.2	90.3	90.3
5000	90.5	90.6	90.9
10000	91.3	91.5	92.0

Tabelle 5.3: Testperplexität, WSJ0–1M und WSJ0–4M, flaches Unigramm–Kriterium, Maximale Abdeckung, für unterschiedliche Phrasenanzahlen P.

P	1M	4M
0	229.7	152.1
1	229.4	151.9
2	229.2	151.9
5	229.2	151.5
10	229.2	151.0
20	229.5	150.5
50	230.0	150.3
100	230.4	150.0
200	232.3	150.3

wobei das hierarchische Unigramm–log–Likelihood–Kriterium die besten Ergebnisse liefert. Im Gegensatz zu den Varigrammen erbringt das Leaving–One–Out–Kriterium überhaupt keine Verbesserung, da grundsätzlich sehr häufige Wortpaare gewählt werden, bei denen der Verlust einer Beobachtung keine Rolle spielt. Das Bigramm–Kriterium erbringt zwar eine Verbesserung im Vergleich zum Unigramm–Kriterium, die jedoch so geringfügig ist, daß sich der erhebliche Mehraufwand nicht lohnt. Eine hierarchische Auswahl mittels Bigramm–Kriterium ist daher gar nicht erst versucht worden, obwohl diese Methode vermutlich nochmals geringfügig besser wäre als das hierarchische Unigramm–Kriterium.

Für den Einbau in den Spracherkenner wurde das akustische Lexikon um die gewählten Wortphrasen erweitert. Daher konnte der Spracherkenner direkt mit dem wortphrasenbasierten Trigramm–Sprachmodell, das aus den Korpora nach der Methode der Maximalen Abdeckung errechnet wurde, arbeiten. Die Notwendigkeit für die Left–to–Right–Inside–Gleichung aus Anhang D entfiel dadurch. Wortfehlerrate und Perplexität wurden automatisch vom Erkenner auf die Wörter zurückgerechnet. Ein erneutes Training des akustischen Modells fand nicht statt.

Tabelle 5.4: Testperplexität, WSJ0–39M, für unterschiedliche Phrasenanzahlen P und verschiedene Auswahlkriterien, Maximale Abdeckung.

P	Unigramm flach	Bigramm flach	L1O–Unigramm flach	Unigramm hierarchisch
0	96.8			
500	91.2	90.6	91.2	90.7
1000	90.5	90.2	90.5	89.9
2000	90.3	90.2	90.3	89.3
5000	90.9	91.7	90.9	89.8
10000	92.0	94.3	92.0	91.2

Für die hierarchische Auswahl auf dem NAB–Korpus stellte sich das gleiche Problem wie bei den Varigrammen, nämlich daß das gesamte Korpus für die Auswahl der besten Phrase vorhanden sein muß, was sich aus Speicherplatzgründen nicht realisieren ließ. Als Abhilfe wurde mit einer Art Batch–Betrieb gearbeitet: Die ersten 40 Wortphrasen wurden flach ausgewählt und dem Vokabular hinzugefügt. Mit diesem Vokabular wurden die nächsten 40 Wortphrasen nach flachem Kriterium bestimmt und dem Vokabular erneut zugefügt. Mit diesem zweimal erweitertem Vokabular wurden dann die restlichen Phrasen ebenfalls flach ausgewählt.

Tabelle 5.5 gibt die Perplexitäten und Wortfehlerraten auf den DEV–Daten des NAB–Korpus für phrasenbasierte Trigramm–Sprachmodelle an. Wie bei den Varigrammen zeigt sich ein deutliches Absinken der Perplexität, wohingegen die Wortfehlerrate schon bei zweihundert Phrasen ein schwaches Minimum erreicht (knapp 2% relative Verbesserung). Wie bei den Varigrammen ist auch hier der Sprachmodell–Skalierungsfaktor kleiner als beim normalen Worttrigramm, was auf ein übertrainiertes Sprachmodell hindeutet. In dieses Bild paßt auch, daß die hierarchische Phrasenauswahl in Tabelle 5.6 zwar die Perplexität weiter verbessert, die Wortfehlerrate aber eher verschlechtert. Somit wurde für die weitere Arbeit lediglich das flache Kriterium verwendet. Tabelle 5.7 gibt dafür das Ergebnis auf den EVL–Daten des NAB–Korpus für die auf den DEV–Daten optimierten Parametern Phrasenanzahl und Skalierungsfaktor an. Es ergibt sich eine Verminderung um 4% in der Perplexität und um gut 1% relativ in der Wortfehlerrate.

Für Verbmobil wurden nochmals verschiedene Auswahlmethoden für Wortphrasen untersucht. Das Ergebnis findet sich in Tabelle 5.8. Für wortphrasenbasierte Trigramm–Sprachmodelle ergibt sich praktisch kein Unterschied durch die Auswahlmethode. Die Tabelle wurde um die Ergebnisse eines phrasenbasierten Bigramm–Sprachmodells ergänzt, um überhaupt Änderungen beobachten zu können. Wie bei den Untersuchungen aus Tabelle 5.4 ergibt sich auch in der Fehlerrate ein ganz geringer Vorsprung bei der Auswahl durch hierarchisches Unigramm–log–Likelihood–Kriterium. Die Ergebnisse wurden auf einem älteren und etwas kleineren Verbmobil–Trainingskorpus durchgeführt und mit dem aktuellen Korpus nicht wiederholt, da die Ergebnisse eindeutig und der Versuch sehr aufwendig ist. Die durch hierarchisches Unigramm–log–Likelihood–Kriterium erzeugten Phrasen sind in Tabelle 5.9 aufgeführt. In Tabelle 5.10 wird das beste Ergebnis auf den

Tabelle 5.5: Testperplexität und Wortfehlerrate, NAB–DEV, nach flachem Unigramm–Kriterium, für unterschiedliche Phrasenanzahlen P.

Modell	P	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Worttrigramm	—	21	121.8	1.8/2.0	12.8
Phrasentrigramm	100	19	119.8	1.7/2.1	12.7
	200	19	119.1	1.7/2.0	12.6
	500	19	117.4	1.8/2.1	12.9
	1000	18	116.7	1.8/2.1	13.0

Tabelle 5.6: Testperplexität und Wortfehlerrate, NAB–DEV, nach hierarchischem Unigramm–Kriterium, für unterschiedliche Phrasenanzahlen P.

Modell	P	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Worttrigramm	—	21	121.8	1.8/2.0	12.8
Phrasentrigramm	100	19	119.0	1.7/2.0	12.7
	200	20	117.9	1.7/2.0	12.7
	500	19	117.0	1.7/2.0	12.8
	1000	17	116.3	1.7/2.1	12.9

Tabelle 5.7: Testperplexität und Wortfehlerrate, NAB–EVL, 200 Phrasen nach flachem Unigramm–Kriterium, Skalierungsfaktor 19.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Worttrigramm	123.4	2.0/2.3	13.6
Phrasentrigramm	118.0	1.9/2.2	13.4

aktuellen Verbmobil–Daten angegeben. In der Fehlerrate ergibt sich im Vergleich zum wortbasierten Trigramm–Sprachmodell eine Verminderung in der Wortfehlerrate um 3% relativ und in der Perplexität ebenfalls um 3%. Da es sich nur um ein sehr kleines Trainingskorpus handelt, fällt dieser Gewinn noch deutlicher aus als in den Ergebnissen aus Tabelle 5.3.

Tabelle 5.8: Testperplexitäten und Wortfehlerraten für wort– und phrasenbasierte Modelle mit unterschiedlichen Auswahlmethoden, Verbmobil (ohne Verwendung der Texte des Datensatzes cd14 im Training).

Auswahlkriterium	Bigramm				Trigramm			
	P	PP	Wortfehler [%]		P	PP	Wortfehler [%]	
			del/ins	WER			del/ins	WER
wortbasiert	—	58.0	4.0/3.5	20.0	—	46.3	3.4/3.2	18.5
Unigramm flach	100	52.9	3.9/3.2	19.2	200	46.1	3.3/3.2	17.8
Unigramm hier.	100	52.7	3.8/3.0	18.8	100	45.6	3.4/3.2	17.9
Bigramm flach	50	52.8	3.7/3.4	19.0	50	45.3	3.3/3.2	17.9

Tabelle 5.9: Beispiele für gewählte Wortpaare, Gewinn in log–Likelihood und Wortpaarhäufigkeiten für hierarchisches Unigramm–log–Likelihood–Kriterium, Verbmobil.

No.	$\Delta F(a,b)$	$N(a,b)$	$a \quad b$
1	5546	1485	bei mir
2	5019	771	mein Name
3	3611	835	einen Termin
4	3314	464	grüß Gott
5	3208	575	guten Tag
6	3177	730	bei Ihnen
7	3144	729	mein_Name ist
8	2792	748	bis zum
9	2635	1058	habe ich
10	2290	832	bin ich
13	2359	530	wie wäre_es
24	1216	291	wie_sieht es
27	1141	237	grüß_Gott (mein_Name)_ist
29	1097	264	wie_(wäre_es) denn
31	1092	416	da habe_ich

Tabelle 5.10: Testperplexität und Wortfehlerrate, Verbmobil, 100 Phrasen nach hierarchischem Unigramm–Kriterium.

	LMSc	PP	Wortfehler [%]	
			del/ins	WER
Worttrigramm	14	40.6	3.3/3.1	17.7
Phrasentrigramm	14	39.5	3.4/3.0	17.2

Zusammenfassung. Zusammenfassend läßt sich feststellen:

- Wortphrasenbasierte Sprachmodelle erbringen zwar eine geringere Verminderung in der Testperplexität als Varigramme, die sich jedoch im Gegensatz zu den Varigrammen auch auf die Wortfehlerrate auswirkt. Weiter entfällt bei den wortphrasenbasierten Sprachmodellen die aufwendige Viergrammsuche im Rescoring.
- Die spezielle Art des log–Likelihood–Kriterium sowie der Berechnung der Wahrscheinlichkeit eines Satzes ist, bezogen auf die Perplexität und Fehlerrate, nicht bedeutend. Die einfachen Methoden können daher ohne spürbare Verluste in der Performanz verwendet werden.

Als Konsequenz werden die wortphrasenbasierten Sprachmodelle als bessere Methode für den verbleibenden Teil dieser Arbeit den Varigrammen vorgezogen. Es wird dabei die Auswahl nach dem hierarchischen Unigramm–Kriterium und die Wahrscheinlichkeitsberechnung nach dem Prinzip der Maximalen Abdeckung erfolgen.

Kapitel 6

Wortklassen

Wie bereits in Kap. 2.2.4 beschrieben, handelt es sich bei wortklassenbasierten Sprachmodellen um Trigramme auf der Basis von Vokabularpartitionen, „Wortklassen“ genannt. Da es weniger Wortklassen als Wörter gibt, vergröbert sich die Modellierung. Gleichzeitig verringert sich die Zahl der Parameter, die sich nun besser schätzen lassen. Die Wortklassen können von Linguisten vorgegeben sein, sogenannte Parts–Of–Speech. Hier jedoch sollen die Wortklassen gebildet werden, die die log–Likelihood der Trainingskorpora maximieren. Für dieses kombinatorische Problem ist keine effiziente Lösung bekannt, deshalb wird ein lokal konvergenter „Cluster–Algorithmus“ basierend auf dem Austausch–Algorithmus [Duda & Hart 73, pp. 227–228] zum Auffinden der geeigneten Wortklassen verwendet. In diesem Kapitel soll Folgendes gezeigt werden:

- Anstatt der bisher in der Literatur verwendeten Bigramm–log–Likelihood als Kriterium für den Cluster–Algorithmus soll die Trigramm–log–Likelihood hierfür untersucht werden.
- Trotz der Beschränkung auf die lokale Konvergenz ist der Cluster–Algorithmus immer noch sehr rechenintensiv. Es soll daher eine effiziente Implementierung entwickelt und ihre genaue Komplexität hergeleitet werden.
- Dieser Cluster–Algorithmus soll auf die drei in dieser Arbeit benutzten Korpora angewandt und seine Laufzeit und seine Auswirkung auf Perplexität und Fehlerrate untersucht werden.

6.1 Wortklassenbasierte Sprachmodelle

Die Partitionierung des Vokabulars in G Wortklassen wird durch die

Abbildungsfunktion $\mathcal{G} : w \rightarrow g_w$

bestimmt, die jedem Wort w seine Wortklasse g_w zuweist. Das Sprachmodell besteht aus zwei Wahrscheinlichkeitsverteilungen [Kneser & Ney 93]:

- einer Transitionswahrscheinlichkeit $p_1(g_w|g_v)$ für ein wortklassenbasiertes Bigramm–Sprachmodell bzw. $p_2(g_w|g_u, g_v)$ für ein wortklassenbasiertes Trigramm–Sprachmodell für die Wahrscheinlichkeit einer Wortklasse gegeben die Wortklassen der Vorgängerwörter und

- einer „Membership Probability“ $p_0(w|g)$ für die Wahrscheinlichkeit der Zugehörigkeit des Wortes w zur Wortklasse g. Wegen $p_0(w|g) = 0$ für $g \neq g_w$ kann auch $p_0(w|g_w)$ geschrieben werden.

Insgesamt ergibt sich

$$p(w|v) = p_0(w|g_w) \cdot p_1(g_w|g_v) \quad \text{bzw.} \tag{6.1}$$
$$p(w|u,v) = p_0(w|g_w) \cdot p_2(g_w|g_u, g_v) \tag{6.2}$$

als wortklassenbasiertes Bigramm– bzw. Trigramm–Sprachmodell. Die Zahl der Parameter hat sich dabei von $W \cdot (W-1)$ auf $G \cdot (G-1)$ bzw. $W^2 \cdot (W-1)$ auf $G^2 \cdot (G-1)$ für die Transitionswahrscheinlichkeit zuzüglich jeweils $(W-G)$ für die Membership Probability verringert.

Für die Bigramm–log–Likelihood ergibt sich somit nach Gl. (6.1):

$$\begin{aligned} F(\mathcal{G}) &= \sum_{v,w} N(v,w) \cdot \log\ p(w|v) \\ &= \sum_{w} N(w) \cdot \log\ p_0(w|g_w) + \sum_{g_v, g_w} N(g_v, g_w) \cdot \log\ p_1(g_w|g_v) \quad . \end{aligned} \tag{6.3}$$

Bei gegebener Abbildungsfunktion $\mathcal{G}$ ergibt sich die Maximum–Likelihood–Parameterschätzung durch Zufügung der Normierungsbedingungen mittels Lagrange–Multiplikatoren und Bilden der Ableitung [Ney et al. 94]. Wie bei den wortbasierten n–gramm–Sprachmodellen erhält man daraus die relativen Häufigkeiten:

$$p_0(w|g_w) = \frac{N(w)}{N(g_w)} \tag{6.4}$$

$$p_1(g_w|g_v) = \frac{N(g_v, g_w)}{N(g_v)} \quad . \tag{6.5}$$

Aus den Schätzungen in Gl. (6.4) und Gl. (6.5) ergibt sich für die log–Likelihood–Gleichung (6.3) [Kneser & Ney 93]:

$$
\begin{aligned}
F(\mathcal{G}) &= \sum_{v,w} N(v,w) \cdot \log p(w|v) \\
&= \sum_{w} N(w) \cdot \log \frac{N(w)}{N(g_w)} + \sum_{g_v,g_w} N(g_v,g_w) \cdot \log \frac{N(g_v,g_w)}{N(g_v)} \\
&= \sum_{g_v,g_w} N(g_v,g_w) \cdot \log N(g_v,g_w) - 2 \cdot \sum_{g} N(g) \cdot \log N(g) \\
&\quad + \sum_{w} N(w) \cdot \log N(w) \quad . \qquad (6.6)
\end{aligned}
$$

Im Rahmen dieser Arbeit wurde auch das entsprechende Trigramm–log–Likelihood–Kriterium hergeleitet, das sich in Anhang G findet.

Bei der Verwendung der wortklassenbasierten Sprachmodelle Gl. (6.1) bzw. Gl (6.2) auf Testkorpora mit ungesehenen Ereignissen werden Transitionswahrscheinlichkeit und Membership Probability separat mit Absolute Discounting geglättet.

6.2 Cluster–Algorithmus

Die unbekannte Abbildungsfunktion $\mathcal{G} : w \rightarrow g_w$ für eine vorgegebene Anzahl Wortklassen G wird mit Hilfe eines Cluster–Algorithmus basierend auf dem lokal konvergenten Austauschverfahren (ISODATA [Duda & Hart 73, pp. 227–228], [Kneser & Ney 93]) ermittelt: Das Vokabular wird Wort für Wort durchlaufen, das aktuelle Wort in jede der G Wortklassen geschoben, die log–Likelihood berechnet und das Wort derjenigen Wortklasse mit der höchsten log–Likelihood zugeordnet. Da sich durch die Zuordnungen gegen Ende des Vokabulars die log–Likelihood für die Wörter am Anfang des Vokabulars ändert, muß das Vokabular in mehreren Iterationen durchlaufen werden. Das Verfahren konvergiert, da es nur eine endliche Anzahl an Aufteilungsmöglichkeiten und damit Abbildungsfunktionen für das Vokabular gibt und gleichzeitig die derzeitige Abbildungsfunktion nur durch solche Abbildungsfunktionen ersetzt werden kann, die die log–Likelihood noch weiter erhöhen. Die initiale Abbildungsfunktion ordnet den häufigsten $G-1$ Wörtern eine eigene Wortklasse und allen übrigen Wörtern die verbleibende Wortklasse zu. Entsprechend wird das Vokabular nach Worthäufigkeit geordnet durchlaufen. Das Abbruchkriterium ist die Konvergenz oder eine vorher festgelegte Anzahl an Iterationen.

Eine Darstellung dieses Algorithmus gibt Abb. 6.1. Die naive Implementierung ist sehr zeitaufwendig: Es gibt die äußere Schleife über die Anzahl der Iterationen I, eine weiter innere Schleife über die Vokabulargröße W und eine innerste Schleife über die Klassenzahl G. Weiter werden für die Berechnung der Bigramm–log–Likelihood Gl. (6.6) G^2 Schritte benötigt. Wird stattdessen die Trigramm–log–Likelihood Gl. (G.3) verwendet, sind es sogar G^3 Schritte. Insgesamt ergibt sich damit ein Aufwand von $I \cdot W \cdot G^3$ bzw. $I \cdot W \cdot G^4$, der bei einer Vokabulargröße W von 20 000 Wörtern und einer Wortklassenzahl G von mehreren hundert Wortklassen nicht mehr zu vertreten ist.

```
start with some initial mapping w → g_w
    for each word w of the vocabulary do
        for each class k do
            tentatively exchange word w from class g_w to class k and update counts
            compute log–likelihood for this tentative exchange
        exchange word w from class g_w to class k with maximum log–likelihood
do until stopping criterion is met
```

Abbildung 6.1: Austauschverfahren zur Bildung von Wortklassen.

```
for each predecessor word v of word w do
    search for N(v, w) in the list of successor words of word v using binary search
    N(g_v, w) := N(g_v, w) + N(v, w)
    put class g_v on the list of predecessor classes of word w
for each successor word x of word w do
    N(w, g_x) := N(w, g_x) + N(w, x)
    put class g_x on the list of successor classes of word w
```

Abbildung 6.2: Berechnung der Hilfszähler $N(g, w)$ und $N(w, g)$.

In dieser Arbeit soll der Aufwand in zwei Schritten für die Bigramm–log–Likelihood rein algorithmisch, d.h. ohne Vereinfachung des log–Likelihood–Kriteriums, deutlich reduziert werden. Dieselben Schritte für die Trigramm–log–Likelihood sind in Anhang G beschrieben. Für diese Schritte werden noch die Hilfszähler

$$N(w, g) = \sum_{x:g_x=g} N(w, x) \tag{6.7}$$

$$N(g, w) = \sum_{v:g_v=g} N(v, w) \tag{6.8}$$

benötigt, die für jedes aktuelle Wort w neu berechnet werden und angeben, wie häufig dieses Wort w vor bzw. nach der Wortklasse g im Trainingskorpus steht. Die Berechnung von Gl. (6.7) erfordert einen Lauf über die Menge $\mathcal{S}(w)$ der Nachfolgerwörter x zu w. Analog erfordert die Berechnung von Gl. (6.8) einen Lauf über die Menge $\mathcal{P}(w)$ der Vorgängerwörter v zu w. Da aus Speicherplatzgründen die Häufigkeiten für das Wortpaar (v, w) nur in der Liste der Nachfolgerwörter zum Wort v abgelegt ist, muß die Häufigkeit für das Wortpaar (v, w) in dieser Liste binär gesucht werden. Die Berechnung ist in Abb. 6.2 dargestellt. Insgesamt ergibt sich also für jedes aktuelle Wort w ein Aufwand von

$$|\mathcal{S}(w)| + \sum_{v \in \mathcal{P}(w)} \log_2(|\mathcal{S}(v)|)$$

für die Erstellung der Hilfszähler in Gl. (6.7) und Gl. (6.8).

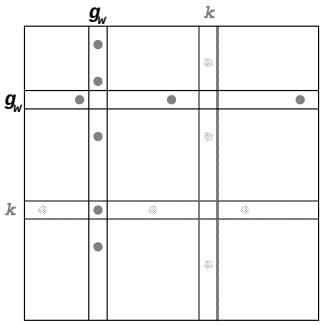

Abbildung 6.3: Schematische Übersicht über die Änderungen in den Wortklassenpaaren durch die Verschiebung eines Wortes von Klasse g_w in die Klasse k.

1. Schritt: Effiziente Berechnung der log–Likelihood

Zur einmaligen Berechnung der Bigramm–log–Likelihood Gl. (6.6) sind zwar G^2 Schritte notwendig, [Ney 94] hat jedoch darauf hingewiesen, daß durch die Verschiebung eines Wortes w von der Wortklasse g_w zu einer anderen Wortklasse k nur diejenigen Summanden betroffen sind, die die Wortklassen g_w oder k enthalten. Wie in Abb. 6.3 gezeigt, sind dies nur $4 \cdot (G-1)$ Terme. Werden nur diese Terme neu berechnet, ist der Aufwand nur noch linear statt quadratisch in der Anzahl der Wortklassen G.

Für die detaillierte Betrachtung wird zwischen der Ausgliederung des Wortes w aus seiner Wortklasse g_w und seiner Eingliederung in die $G-1$ übrigen Wortklassen unterschieden und die Ausgliederung genauer betrachtet. Die Herleitung für die Eingliederung ist analog. Zuerst wird die Häufigkeit der Wortklasse g_w um die Häufigkeit von w vermindert:

$$N(g_w) := N(g_w) - N(w) \quad .$$

Dann wird für eine Wortklasse $g \neq g_w$ die Häufigkeit der Übergänge von g nach g_w bzw. von g_w nach g um die Anzahl der Übergänge von g nach w bzw. von w nach g_w vermindert:

$$\forall g \neq g_w : N(g, g_w) := N(g, g_w) - N(g, w) \tag{6.9}$$

$$\forall g \neq g_w : N(g_w, g) := N(g_w, g) - N(w, g) \quad . \tag{6.10}$$

Die Häufigkeit des Übergangs von g_w nach g_w ist etwas komplizierter, da sowohl die Häufigkeit des Übergangs von g_w nach w (Gl. (6.9)) als auch von w nach g_w (Gl. (6.10)) abgezogen werden muß und der Übergang von w nach w selbst in beiden abzuziehenden Häufigkeiten enthalten ist. Zum Ausgleich muß er einmal wieder aufaddiert werden:

$$N(g_w, g_w) := N(g_w, g_w) - N(g_w, w) - N(w, g_w) + N(w, w) \quad . \tag{6.11}$$

Gl. (6.11) ist eine Anwendung der Siebformel [Engesser 94, S. 561], siehe dazu auch Abb. 6.4. Schließlich müssen noch die Hilfszähler $N(g_w, w)$ und $N(w, g_w)$ angepaßt werden:

$$N(g_w, w) := N(g_w, w) - N(w, w)$$
$$N(w, g_w) := N(w, g_w) - N(w, w) \quad .$$

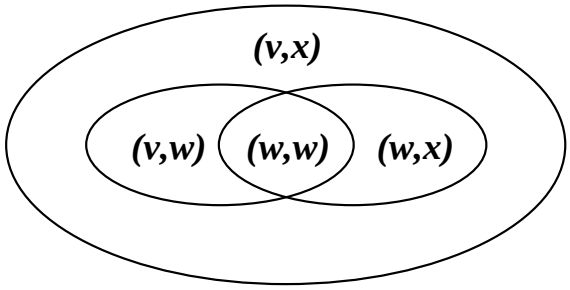

Abbildung 6.4: Siebformel angewandt auf die Ausgliederung eines Wortes w aus seiner Wortklasse g_w. (v, x) ist ein beliebiges Wortpaar mit $v,\ x \in g_w$, (v, w) ist ein beliebiges Wortpaar mit $v \in g_w$, und (w, x) ist ein beliebiges Wortpaar mit $x \in g_w$.

Die effiziente Berechnung der Änderung in der Bigramm–log–Likelihood durch Ausgliederung eines Wortes w aus seiner Klasse g_w ist in Abb. 6.5 dargestellt. Der Aufwand beträgt etwa $G + G$ Schritte und ist für die Eingliederung des Wortes w in eine Klasse k genauso groß. Anstatt also für G Wortklassen die Bigramm–log–Likelihood in jeweils etwa G^2 Schritten zu berechnen, sind nun nur noch eine Ausgliederung mit $G+G$ Schritten und $G-1$ Eingliederungen mit $G+G$ Schritten durchzuführen. Insgesamt ergibt sich also für den in Abb. 6.1 dargestellten Cluster–Algorithmus mit der in Abb. 6.5 dargestellten effizienten Berechnung der Änderung der Bigramm–log–Likelihood und der Berechnung der Hilfszähler in Abb. 6.2 ein Aufwand von

$$I \cdot \sum_w \left(|\mathcal{S}(w)| + \sum_{v \in \mathcal{P}(w)} \log_2(|\mathcal{S}(v)|) + G \cdot (G + G) \right) \quad .$$

Mit B als Anzahl unterschiedlicher Wort–Bigramme im Trainingskorpus gilt

$$|\mathcal{P}(w)| \cdot \log_2 \left(\frac{B}{W} \right) \geq \sum_{v \in \mathcal{P}(w)} \log_2(|\mathcal{S}(v)|) \quad , \tag{6.12}$$

$$\sum_w |\mathcal{S}(w)| = \sum_w |\mathcal{P}(w)| = B \quad ,$$

und somit als Schätzung für die obere Schranke des Berechnungsaufwands

$$I \cdot \left(B + B \cdot \log_2 \left(\frac{B}{W} \right) + 2 \cdot W \cdot G^2 \right) \quad . \tag{6.13}$$

2. Schritt: Benutzung positiver Hilfszähler

Gl. (6.9) und Gl. (6.10) gelten für alle Wortklassen $g \neq g_w$, jedoch zeigt sich nur ein Effekt für die Hilfszähler $N(g, w) \neq 0$ und $N(w, g) \neq 0$. Zu einem Wort w kann es nicht mehr Vorgänger– bzw. Nachfolgerwortklassen geben als Vorgänger– und Nachfolgerwörter und von diesen nicht mehr als die Häufigkeit des Wortes w im Trainingskorpus. Da die meisten Wörter eines Vokabulars eine eher geringe Häufigkeit haben, liegt die Vermutung

for each class $g \neq g_w$ do
$N(g, g_w) := N(g, g_w) - N(g, w)$
for each class $g \neq g_w$ do
$N(g_w, g) := N(g_w, g) - N(w, g)$
$N(g_w) := N(g_w) - N(w)$
$N(g_w, g_w) := N(g_w, g_w) - N(g_w, w) - N(w, g_w) + N(w, w)$
$N(g_w, w) := N(g_w, w) - N(w, w)$
$N(w, g_w) := N(w, g_w) - N(w, w)$

Abbildung 6.5: Effiziente Berechnung der Änderung in der Bigramm–log–Likelihood durch die Ausgliederung eines Wortes w aus seiner Klasse g_w.

nahe und wird später auch experimentell bestätigt, daß schon ab Wortklassenzahlen G von 100 und darüber im Mittel deutlich weniger Nachfolger– bzw. Vorgängerwortklassen für die Wörter des Vokabulars existieren als Wortklassen insgesamt. Damit würde bei einer naiven Implementierung der Gl. (6.9) und Gl. (6.10), die sich in der innersten Schleife des Algorithmus befinden, sehr viel Rechenzeit für Rechenoperationen ohne Effekt verwendet. Es ist daher effizienter, bei der Erstellung der Hilfszähler $N(g, w)$ und $N(w, g)$ in Abb. 6.2 den Mehraufwand einer Liste der Hilfszähler $N(g, w),\ N(w, g) > 0$ zu betreiben und nur diese in den Gl. (6.9) und Gl. (6.10) zu benutzen. Diese Verfeinerung ist in Abb. 6.6 dargestellt.

Bezeichne $G(w, \cdot)$ die Anzahl der Wortklassen g mit $N(w, g) > 0$, $G(\cdot, w)$ die Anzahl der Wortklassen g mit $N(g, w) > 0$ und $\bar{G}_{w\cdot}$ und $\bar{G}_{\cdot w}$ die jeweiligen Mittelwerte

$$\bar{G}_{w\cdot} = \frac{1}{W} \cdot \sum_{w} G(w, \cdot)$$
$$\bar{G}_{\cdot w} = \frac{1}{W} \cdot \sum_{w} G(\cdot, w) \quad .$$

Dann ergibt sich für den Cluster–Algorithmus mit dieser Verfeinerung eine Aufwandsabschätzung von

$$I \cdot \left(B + B \cdot \log_2 \left(\frac{B}{W} \right) + \left(\sum_{w} G \cdot (G(w, \cdot) + G(\cdot, w)) \right) \right)$$
$$= I \cdot \left(B + B \cdot \log_2 \left(\frac{B}{W} \right) + W \cdot G \cdot (\bar{G}_{w\cdot} + \bar{G}_{\cdot w}) \right) \quad . \tag{6.14}$$

6.3 Experimentelle Ergebnisse

Der Cluster–Algorithmus wurde wie in Kap. 6.2 für das Bigramm–log–Likelihood–Kriterium und in Anhang G für das Trigramm–log–Likelihood–Kriterium beschrieben implementiert. Sonderzeichen, z.B. für Satzende und unbekanntes Wort, wurden separat

for each *seen* predecessor class $g \neq g_w$ of word w do
$N(g, g_w) := N(g, g_w) - N(g, w)$
for each *seen* successor class $g \neq g_w$ of word w do
$N(g_w, g) := N(g_w, g) - N(w, g)$
$N(g_w) := N(g_w) - N(w)$
$N(g_w, g_w) := N(g_w, g_w) - N(g_w, w) - N(w, g_w) + N(w, w)$
$N(g_w, w) := N(g_w, w) - N(w, w)$
$N(w, g_w) := N(w, g_w) - N(w, w)$

Abbildung 6.6: Verfeinerte effiziente Berechnung der Änderung in der Bigramm–log–Likelihood aufgrund von Listen der positiven Hilfszähler.

in zusätzlichen Einzelwortklassen abgelegt und nicht in den Cluster–Algorithmus mit einbezogen. Für das Bigramm–log–Likelihood–Kriterium wurde der Cluster–Algorithmus ausiteriert, für das Trigramm–log–Likelihood–Kriterium aus Laufzeitgründen nach 10 Iterationen abgebrochen, da keine bedeutenden Veränderungen mehr zu erwarten waren. Auch aus Laufzeitgründen konnten für das Trigramm–log–Likelihood–Kriterium nur Wortklassenzahlen G von 50, 100 und 200 getestet werden, für das Bigramm–log–Likelihood–Kriterium hingegen zusätzlich noch 500, 1000 und 2000.

Die Laufzeiten für beide Kriterien und die verschiedenen Klassenzahlen G finden sich in Tab. 6.1. Die Laufzeiten in Abhängigkeit von der Korpusgröße steigen in etwa linear mit der Anzahl der Bi– bzw. Trigramme, wie nach Gl. (6.14) bzw. Gl. (G.5) zu erwarten. Die Laufzeiten in Abhängigkeit von der Klassenzahl G steigen schneller als linear, jedoch als Konsequenz der verfeinerten Implementierung deutlich langsamer als quadratisch bzw. kubisch. Die Anzahl Iterationen für das Bigramm–log–Likelihood–Kriterium liegt zwischen 14 und 32 ohne offensichtlichen Bezug zur Korpusgröße oder Klassenzahl.

Um den Gewinn durch die verfeinerte Implementierung nachzuweisen, wurde für eine Klassenzahl $G = 500$ und das Bigramm–log–Likelihood–Kriterium ein Cluster–Algorithmus mit einer Berechnung der Änderung in der log–Likelihood nach Abb. 6.5 statt Abb. 6.6 getestet. Das Ergebnis in Tab. 6.2 ist von der Korpusgröße nahezu unabhängig und zeigt die Dominanz des Terms $2 \cdot W \cdot G^2$ für die Wortverschiebung aus Gl. (6.13). Bei der Verfeinerung ist hingegen nur der Aufwand $W \cdot G \cdot (\bar{G}_{\cdot w} + \bar{G}_{w\cdot})$ nötig. Auf WSJ0–39M wurde $\bar{G}_{\cdot w} \approx 81$ und $\bar{G}_{w\cdot} \approx 86$ gemessen. Durch die Verfeinerung ergibt sich dadurch eine Beschleunigung um den Faktor vier oder mehr im Vergleich zu Tab. 6.1.

Tab. 6.3 zeigt Beispiele aus $G = 100$ Wortklassen für das WSJ0–39M–Korpus. Für eine repräsentative Auswahl sind die häufigsten Wörter jeder zehnten Wortklasse angegeben. Die meisten Wortklassen haben eine syntaktische Bedeutung, wie z.B. Nomen im Genitiv (Wortklasse $g = 2$) oder Adjektive (Wortklasse $g = 42$). Manchmal gibt es auch einen semantischen Zusammenhang wie etwa für Wortklasse $g = 12$ mit Verben für kommunikative Tätigkeiten. Allerdings gibt es fast immer unpassende Zuordnungen oder sogar

Tabelle 6.1: CPU–Sekunden pro Iteration auf einer SGI Workstation mit R4400–Prozessor, WSJ0.

G	Bigramm–Kriterium			Trigramm–Kriterium		
	1M	4M	39M	1M	4M	39M
50	14	29	91	232	481	1635
100	28	63	183	682	1512	5005
200	120	139	399	2577	5790	21534
500	167	430	1563	-		
1000	449	1675	3971	-		
2000	1097	2905	10074	-		

Tabelle 6.2: CPU–Sekunden pro Iteration für das Bigramm–log–Likelihood–Kriterium auf einer SGI Workstation mit R4400–Prozessor, WSJ0, ohne Liste der positiven Hilfszähler.

G	1M	4M	39M
500	6270	7206	7686

sehr heterogene Wortklassen wie Wortklasse $g = 32$ mit Verben in unterschiedlichen Zeitformen.

Die Perplexitäten auf den Trainingskorpora in Tab. 6.4 zeigen eine umso bessere Beschreibung der Trainingstexte durch das wortklassenbasierte Bigramm– bzw. Trigramm–Sprachmodell, je umfangreicher mit steigender Wortklassenzahl G und steigender Historienlänge (von Bigramm auf Trigramm) das Sprachmodell und je kleiner und damit einfacher beschreibbarer das Trainingskorpus ist. Auch ist klar die Vergröberung durch die Wortklassen im Vergleich zum wortbasierten Bigramm– bzw. Trigramm–Sprachmodell (entspricht $G = 19979$ zzgl. den beiden Zusatzklassen für Satzende und unbekanntes Wort) zu sehen.

Für die Berechnung der Testperplexitäten wurden wortklassenbasierte Bigramm–Sprachmodelle für die Bigramm–log–Likelihood–trainierten Wortklassen und wortklassenbasierte Trigramm–Sprachmodelle für die Trigramm–log–Likelihood–trainierten Wortklassen gebildet. Da die Wortklassenbildung für das Trigramm–log–Likelihood–Kriterium nur bis $G = 200$ Wortklassen möglich war, wurden zusätzlich wortklassenbasierte Trigramm–Sprachmodelle für die Bigramm–log–Likelihood–trainierten Wortklassen gebildet. Die Testperplexitäten stehen in Tab. 6.5. Folgende Schlüsse lassen sich ziehen:

- Der Nachteil der Vergröberung des Sprachmodells wird nicht durch die bessere Schätzung der Modellparameter ausgeglichen. Zwar sinken die Perplexitäten mit

Tabelle 6.3: Jede zehnte aus $G = 100$ mit Trigramm–log–Likelihood–Kriterium gebildeten Wortklassen auf dem WSJ0–39M–Korpus.

$g = 2$	THE, JAPAN'S, YESTERDAY'S, BRITAIN'S, TODAY'S, CANADA'S, CHINA'S, FRANCE'S, MEXICO'S, ITALY'S, AUSTRALIA'S, ISRAEL'S, CALIFORNIA'S, TOKYO'S, TAIWAN'S, NICARAGUA'S, SWEDEN'S, POLAND'S, NASDAQ'S, TOMORROW'S, ...
$g = 12$	SAID, SAYS, ADDS, SUCCEEDS, CONTENDS, RECALLS, EXPLAINS, ASKS, PREDICTS, CONCEDES, SUCCEEDING, INSISTS, ASSERTS, WARNS, ADMITS, COMPLAINS, REPLIED, CONCLUDES, DECLARES, OBSERVES, ...
$g = 22$	BY, THEREBY
$g = 32$	PLANS, AGREED, EXPECTS, BEGAN, DID, MAKES, CAME, TOOK, GOT, DOES, CONTINUED, CALLS, HELPED, WANTS, DECIDED, WENT, MEANS, OWNS, FAILED, HOLDS, ...
$g = 42$	NEW, MAJOR, BIG, OLD, FULL, ADDITIONAL, SINGLE, NON, JOINT, LEADING, WIDE, DOUBLE, LEVERAGED, PRE, PARTICULAR, CONVENTIONAL, TRIPLE, COMPARABLE, FORT, GRAMM, ...
$g = 52$	U., JONES, BROTHERS, LYNCH, LEHMAN, STANLEY, HUTTON, SACHS, REYNOLDS, BACHE, PEABODY, INDUSTRIALS, STEARNS, HANOVER, WITTER, GENERALE, KRAVIS, LUFKIN, GUARANTY, GRENFELL, ...
$g = 62$	THAN, QUARTER, HALF, EIGHTHS, QUARTERS, EIGHTH, SIXTEENTHS, INTERSTATE'S
$g = 72$	BUSINESS, INTEREST, TAX, TRADE, DEBT, MONEY, CAPITAL, MANAGEMENT, WORK, CASH, GROWTH, PRODUCTION, POLICY, POWER, NEWS, CREDIT, TAKEOVER, SUPPORT, BUDGET, INFORMATION, ...
$g = 82$	INCORPORATED, CORPORATION, GROUP, UNIT, LIMITED, MAKER, INDUSTRIES, DIVISION, UNIVERSITY, HOLDINGS, SUBSIDIARY, PARTNERSHIP, CORPORATION'S, ASSOCIATES, INCORPORATED'S, OPERATOR, BANCORP, AFFILIATE, SUPPLIER, LABORATORIES, ...
$g = 92$	OFFICIALS, IT'S, ANALYSTS, TRADERS, EXECUTIVES, THAT'S, WE'RE, SOURCES, THERE'S, DEALERS, BANKERS, I'M, THEY'RE, HE'S, ECONOMISTS, BROKERS, STATISTICS, YOU'RE, EXPERTS, CRITICS, ...

Tabelle 6.4: Perplexitäten für wortklassenbasierte Bigramm– und Trigramm–Sprachmodelle auf den WSJ0–Trainingskorpora.

G	Bigramm			Trigramm		
	1M	4M	39M	1M	4M	39M
50	397.1	415.0	426.2	309.8	329.3	348.7
100	333.1	347.7	357.2	200.0	245.0	266.2
200	280.0	295.3	307.0	94.7	152.9	194.2
500	203.2	233.7	248.0	-		
1000	150.5	188.9	211.4	-		
2000	110.2	150.2	179.7	-		
19979	66.1	95.1	126.5	6.9	12.1	23.3

steigender Klassenzahl und Korpuslänge, jedoch werden die Testperplexitäten für das wortbasierte Bigramm– und Trigramm–Sprachmodell, die in Tab. 6.6 wiederholt sind, nicht erreicht.

- Die auf WSJ0–1M mit dem Trigramm–log–Likelihood–Kriterium trainierten klassenbasierten Sprachmodelle steigen sogar in der Testperplexität mit steigender Klassenzahl G. Offensichtlich ist dieses Korpus zu klein für eine gute statistische Grundlage zur Auswahl der Wortklassen, so daß eine Überanpassung des klassenbasierten Trigramm–Sprachmodells an das Trainingskorpus stattfindet.

- Aus dem vermutlich selben Grund ist das wortklassenbasierte Trigramm–Sprachmodell mit Trigramm–log–Likelihood–trainierten Wortklassen bei $G = 200$ Wortklassen etwas schlechter als mit den Bigramm–log–Likelihood–trainierten Wortklassen. Insgesamt ergibt sich durch das Trigramm–log–Likelihood–Kriterium kaum eine Verbesserung im Vergleich zum Bigramm–log–Likelihood–Kriterium, so daß sich der Mehraufwand dafür nicht lohnt.

Um sicherzugehen, daß die schlechten Testperplexitäten nicht von einer falschen initialen Abbildungsfunktion $\mathcal{G} : w \rightarrow g_w$ herrühren, wurden verschiedene initiale Abbildungsfunktionen getestet mit dem Ergebnis, daß die initiale Abbildungsfunktion kaum eine Auswirkung auf die resultierende Testperplexität hat. Dies ist in Anhang H beschrieben. Eine Erklärung dafür ist die Beobachtung, daß der Verlauf des Cluster–Algorithmus von den häufigsten Wörtern dominiert wird. Die Vergröberung des Modells durch Zusammenlegung von Wörtern in Wortklassen betrifft die häufigsten Wörter am meisten, deshalb werden die häufigsten Wörter vom log–Likelihood–Kriterium möglichst separaten Wortklassen zugeordnet, unabhängig von der initialen Abbildungsfunktion. Die Konsequenz davon ist aber auch, daß es mit diesem log–Likelihood–gesteuerten Cluster–Algorithmus niemals eine geschlossene Wortklasse häufiger Wörter, z.B. der Ziffern oder Präpositionen, geben wird.

Tabelle 6.5: Perplexitäten für wortklassenbasierte Bigramm– und Trigramm–Sprachmodelle auf dem WSJ0–Testkorpus.

G	Bigramm			Trigramm			Trigramm (Bigramm–Kriterium)		
	1M	4M	39M	1M	4M	39M	1M	4M	39M
50	454.5	427.8	421.3	396.5	347.1	343.4	409.7	373.7	361.2
100	396.9	361.1	352.7	415.9	285.7	264.9	352.9	295.9	278.5
200	360.4	310.8	301.8	479.9	258.9	206.2	321.9	245.8	218.5
500	326.8	259.9	244.2	-			286.2	199.7	160.5
1000	318.1	233.5	211.0	-			274.8	176.5	132.2
2000	305.9	218.6	187.1	-			263.1	164.2	113.8

Tabelle 6.6: Perplexitäten für wortbasierte Bigramm– und Trigramm–Sprachmodelle auf dem WSJ0–Testkorpus.

	1M	4M	39M
Bigramm	272.5	205.4	162.5
Trigramm	229.7	152.1	96.8

In einem weiteren Versuch wurden die wortklassenbasierten Sprachmodelle dem wortbasierten Trigramm–Sprachmodell $p_W(w|u,v)$ hinzuinterpoliert:

$$p(w|u,v) = \lambda \cdot p_W(w|u,v) + (1-\lambda) \cdot p_0(w|g_w) \cdot p_1(g_w|g_v) \quad \text{bzw.} \tag{6.15}$$

$$p(w|u,v) = \lambda \cdot p_W(w|u,v) + (1-\lambda) \cdot p_0(w|g_w) \cdot p_2(g_w|g_u,g_v) \quad . \tag{6.16}$$

Damit dienen die wortklassenbasierten Sprachmodelle dem wortbasierten Trigramm–Sprachmodell als zusätzliche Glättung. Da im Interpolationsmodell sowohl die wortklassenbasierten Sprachmodelle als auch das wortbasierte Trigramm–Sprachmodell als Spezialfall für den Interpolationsparameter $\lambda = 0$ bzw. $\lambda = 1$ enthalten sind, kann bei optimal gewähltem Interpolationsparameter λ das Interpolationsmodell niemals schlechter sein als das beste der beiden Teilmodelle. In den Experimenten wurde der Interpolationsparameter λ mittels vereinfachter Kreuzvalidierung auf den Trainingskorpora bestimmt.

Tab. 6.7 zeigt die Testperplexitäten sowie die kreuzvalidierten Interpolationsfaktoren für die Interpolationsmodelle Gl. (6.15) und Gl. (6.16). Die Optima liegen bei relativ geringen Wortklassenzahlen G, da sich für große Wortklassenzahlen G die wortklassenbasierten Sprachmodelle den wortbasierten Sprachmodellen immer mehr annähern und folglich zwei sehr ähnliche Modelle interpoliert werden, mit zu erwartenden geringen Effekten. Die Performanz der wortklassenbasierten Trigramm–Sprachmodelle mit Bigramm– bzw. Trigramm–log–Likelihood–bestimmten Wortklassen unterscheidet sich auch hier kaum, so daß sich wiederum der Aufwand für das Trigramm–log–Likelihood–Kriterium nicht lohnt. Auf dem WSJ0–1M–Korpus sind die Testperplexitäten aufgrund der bereits in

Tabelle 6.7: Interpolation von wortklassenbasierten Bigramm– und Trigramm–Sprachmodellen mit dem wortbasierten Trigramm–Sprachmodell, WSJ0: a) Testperplexitäten; b) Interpolationsparameter λ.

	G	Bigramm			Trigramm			Trigramm (Bigramm–Kriterium)		
		1M	4M	39M	1M	4M	39M	1M	4M	39M
a)	50	214.4	147.2	96.4	205.7	142.9	95.8	208.7	144.8	96.0
	100	210.5	145.4	96.3	211.2	139.7	94.8	203.2	140.8	95.0
	200	208.4	143.9	95.9	225.3	140.1	93.7	202.4	138.1	94.0
	500	208.9	142.0	95.5	-			206.5	137.1	92.9
	1000	213.0	141.7	95.3	-			214.5	138.5	92.4
	2000	218.6	143.1	95.1	-			225.0	142.5	92.8
b)	50	0.70	0.85	0.95	0.65	0.80	0.90	0.65	0.80	0.90
	100	0.65	0.80	0.90	0.65	0.75	0.90	0.60	0.75	0.90
	200	0.60	0.80	0.90	0.75	0.75	0.85	0.60	0.70	0.85
	500	0.55	0.75	0.90	-			0.55	0.65	0.80
	1000	0.55	0.70	0.85	-			0.50	0.65	0.75
	2000	0.55	0.70	0.85	-			0.40	0.60	0.70

Tab. 6.5 ersichtlichen schlechten Wortklassenbildung deutlich schlechter. Den deutlichsten Gewinn mit etwa 12% relativ gibt es, da hier auch der größte Bedarf für Glättung besteht, auf dem kleinen WSJ0–1M–Korpus bei $G = 200$ Wortklassen für wortklassenbasiertes Trigramm–Sprachmodell mit Bigramm–log–Likelihood–trainierten Wortklassen. Auf den größeren Korpora sind die Reduktionen in der Perplexität nicht so deutlich.

Die Rescoring–Experimente auf dem NAB–Korpus zeigen ein ähnliches Bild wie die Ergebnisse für WSJ0. In Tabelle 6.8 bleiben sowohl die Perplexitäten als auch die Wortfehlerrate für reine Klassenmodelle unter der Performanz des wortbasierten Trigramm–Sprachmodells. Die Interpolation der beiden Modelle erbringt auf dem NAB–DEV–Korpus eine Verbesserung beim Optimum von $G = 2000$ Wortklassen um 5% in der Perplexität und um 3% relativ in der Wortfehlerrate. Geht man mit diesen optimalen Parametern auf das NAB–EVL–Korpus, so bleibt nach Tabelle 6.9 immer noch eine Verbesserung von 4% in der Perplexität und 2% relativ in der Wortfehlerrate.

Die Interpolation von Wortklassen zum einfachen Trigramm–Sprachmodell erzielt somit eine merkliche Verbesserung in Perplexität und Fehlerrate. In einem weiteren Experiment wurde untersucht, wieweit sich ein phrasenbasiertes Trigramm–Modell als bisher bestes Sprachmodell durch die Interpolation mit Wortklassen verbessern läßt. Die Ergebnisse finden sich in den Tabellen 6.10 und 6.11. Auf dem NAB–DEV–Korpus lassen sich immer noch Verbesserungen von 4% in der Perplexität und 3% in der Wortfehlerrate erzielen. Auf dem NAB–EVL–Korpus erreicht man mit diesen Parametern zwar immer noch eine Verminderung in der Perplexität um 4%, jedoch in der Wortfehlerrate nur um 1% relativ. Da die Perplexität auf beiden NAB–Korpora gleich stark sinkt, kann die Ursache dafür

Tabelle 6.8: Testperplexität und Wortfehlerrate, NAB–DEV, wortbasiertes und klassenbasiertes Trigramm–Sprachmodell, Wortklassenbildung nach Bigramm–Kriterium, für unterschiedliche Klassenanzahlen G.

Modell	G	λ	LMSc	PP	Wortfehler [%]	
					del/ins	WER
Worttrigramm	—	—	21	121.8	1.8/2.0	12.8
Klassentrigramm	1000	—	19	168.1	2.2/2.3	14.9
	2000	—	17	144.5	1.9/2.3	13.8
	5000	—	20	128.9	1.8/2.1	13.0
Worttrigramm	1000	0.8	18	116.3	1.7/2.1	12.6
+ Klassentrigramm	2000	0.7	21	116.7	1.8/2.1	12.4
	5000	0.7	20	119.0	1.7/2.1	12.5

Tabelle 6.9: Testperplexität und Wortfehlerrate, NAB–EVL, Wortklassen nach Bigramm–Kriterium, Wortklassenanzahl G, Interpolationsparameter λ und Skalierungsfaktor optimiert auf dem NAB–DEV–Korpus.

Modell	G	λ	LMSc	PP	Wortfehler [%]	
					del/ins	WER
Worttrigramm	—	—	21	123.4	2.0/2.3	13.6
Klassentrigramm	5000	—	20	130.6	2.2/2.4	14.0
Worttrigramm + Klassentrigramm	2000	0.7	21	118.7	2.0/2.2	13.3

eigentlich nur in den Parametern des akustischen Modells liegen, die ja ebenfalls nur auf dem NAB–DEV–Korpus optimiert wurden.

Tab. 6.12 zeigt das Ergebnis des Cluster–Algorithmus auf dem wortphrasenbasierten Verbmobil–Korpus für $G = 100$ Wortklassen zzgl. acht Zusatzklassen für die Sonderzeichen Satzende, unbekanntes Wort, Häsitationen etc. Das verhältnismäßig kleine Korpus bietet nur eine schwache statistische Grundlage für die Wortklassenbildung. Die häufigeren Wörter und Wortphrasen passen meistens recht gut zueinander, aber es gibt fast immer gänzlich unpassende Zuordnungen seltener Wörter, wie beispielsweise in Wortklasse $g = 38$ zu sehen ist. Bei sich ähnlichen häufigen Wörtern entstehen so homogen erscheinende Wortklassen, wie z.B. $g = 8$, $g = 18$, $g = 68$ und $g = 98$ neben sehr heterogenen Wortklassen wie z.B. $g = 48$, $g = 58$ und $g = 78$.

Trotz dieser zum Teil sehr heterogenen Wortklassen erzielt in Tabelle 6.13 das wortklassenbasierte Trigramm–Modell alleine dieselbe Wortfehlerrate wie das wortbasierte. Dies ist aber offenbar weniger ein Zeichen für die Qualität des wortklassenbasierten Sprachmodells als vielmehr eines dafür, wie schlecht bereits das wortbasierte Trigramm–Sprachmodell auf dem kleinen Verbmobil–Korpus trainiert ist. Die Interpolation der

Tabelle 6.10: Testperplexität und Wortfehlerrate, NAB–DEV, 200 Wortphrasen, wortbasiertes und klassenbasiertes Trigramm–Sprachmodell, Wortklassenbildung nach Bigramm–Kriterium, für beste Klassenanzahl G.

Modell	G	λ	LMSc	PP	Wortfehler [%]	
					del/ins	WER
Phrasentrigramm	—	—	19	119.1	1.7/2.0	12.6
+ Klassentrigramm	2000	0.8	21	114.2	1.6/1.9	12.2

Tabelle 6.11: Testperplexität und Wortfehlerrate, NAB–EVL, 200 Wortphrasen, Wortklassen nach Bigramm–Kriterium, Wortklassenanzahl G, Interpolationsparameter λ und Skalierungsfaktor optimiert auf dem NAB–DEV–Korpus.

Modell	G	λ	LMSc	PP	Wortfehler [%]	
					del/ins	WER
Phrasentrigramm	—	—	19	118.0	1.9/2.2	13.4
+ Klassentrigramm	2000	0.8	21	112.9	2.0/2.2	13.3

beiden Modelle erbringt eine relative Verbesserung von 2% auf dem wortbasierten und 4% auf dem wortphrasenbasierten Korpus. Die Interpolationsparameter wurden mit vereinfachter Kreuzvalidierung auf den Trainingskorpora bestimmt. Bei diesen kleinen Trainingkorpora ist diese Methode allerdings problematisch: Wird der Interpolationsparameter direkt auf dem Testkorpus bestimmt, so wird eine bessere Perplexität von 37.5 und eine deutlich bessere Fehlerrate von 16.8% auf dem wortbasierten Korpus erzielt. Es findet also eine Überanpassung an den Held–out–Teil statt.

Zusammenfassung. In diesem Kapitel wurde ein effizienter Algorithmus zur Bestimmung von Wortklassen vorgestellt. Der Algorithmus basiert auf einem lokal konvergenten Austauschverfahren mit der log–Likelihood als zu optimierendem Kriterium. Dabei wurde erstmals die Trigramm–log–Likelihood statt der bisher üblichen Bigramm–log–Likelihood verwendet, zwei Ansätze zur Beschleunigung des Algorithmus beschrieben und der jeweilige Aufwand abgeschätzt.

Mit diesem effizienten Algorithmus konnten auf einer üblichen Workstation innerhalb weniger CPU–Stunden für 2000 Wortklassen nach Bigramm–log–Likelihood–Kriterium und innerhalb weniger CPU–Tage für 200 Wortklassen nach Trigramm–log–Likelihood–Kriterium Wortklassen gefunden werden, die tatsächlich syntaktische und semantische Zusammenhänge erkennen lassen, allerdings mit einigen unpassenden Zuordnungen. Wortklassenbasierte Sprachmodelle konnten nicht oder nur höchstens die Perplexität und Wortfehlerrate der wortbasierten Sprachmodelle erreichen, wobei sich keine wesentliche Verbesserung durch das Trigramm–log–Likelihood–Kriterium ergeben hat. Die Interpolation beider Modelle erbringt aber eine Verringerung in der Perplexität um bis zu 12%

Tabelle 6.12: Jede zehnte aus $G = 100$ mit Bigramm–log–Likelihood–Kriterium gebildeten Wortklassen auf dem wortphrasenbasierten Verbmobil–Korpus.

$g = 8$	ja, oh, ach, na, tja, nun, ah, aha, jawohl, nja, ...
$g = 18$	der, des, meiner, Ihr, eines, Ihrer, meines, dein, kommender, kurzer, ...
$g = 28$	im, Anfang, Ende, Mitte, Wechsel, Wochenanfang, achtundzwanzigstem, anfangs, aufzuzählen, einundzwanzigstem, ...
$g = 38$	geht_es, wäre_es, haben_Sie, würde_es, hätten_Sie, ginge_es, schaut_es, üblich, schwebt, Tai, ...
$g = 48$	zu, Emil, drüber, miteinander, dorthin, anwesend, genannt, Drei, Ida, Theodor, ...
$g = 58$	hier, in_der_Filiale, mein_Name, freut, verbunden, ehrlich, groß, günstiger, meinerseits, geschrieben, ...
$g = 68$	können, könnten, haben, sind, sollten, müssen, müßten, sehen, wollen, lassen, ...
$g = 78$	wenn, sagten, hast, schlagen, warum, falls, warten, sähe, grüß, entschuldigen, ...
$g = 88$	was, was_halten_Sie, viel, kurz, schnell, lange, weiter, lang, weit, wenig, ...
$g = 98$	Gürtner, Quell, Walberg, Flex, Köpp, Jänsch, Niehmeyer, von_Sudniz, Müller, Schlizio, ...

Tabelle 6.13: Testperplexitäten und Wortfehlerraten für wortbasiertes und wortklassenbasiertes Trigramm–Sprachmodell, Verbmobil.

Trainingstext	Modell	G	LMSc	PP	Wortfehler [%]	
					del/ins	WER
wortbasiert	Worttrigramm	—	14	40.6	3.3/3.1	17.7
	Klassentrigramm	500	16	41.8	3.6/3.0	17.7
	Worttrigramm + Klassentrigramm	100	17	37.9	3.7/2.7	17.3
phrasenbasiert	Worttrigramm	—	14	39.5	3.4/3.0	17.2
	Klassentrigramm	1000	17	40.0	3.5/2.7	17.2
	Worttrigramm + Klassentrigramm	100	18	36.2	3.7/2.6	16.5

und in der Wortfehlerrate um bis zu 4% relativ. Wortklassen sind damit eine sinnvolle Ergänzung zum wortbasierten Trigramm–Sprachmodell.

Kapitel 7

Abstand–m–gramme

Ein Sprachmodell basierend auf Abstand–2–Trigrammen und Trigrammen $p(w|u,v)$, $p(w|t,\cdot,v)$ und $p(w|t,u,\cdot)$ deckt denselben Bereich ab wie ein Viergramm (siehe Abb. 2.6). Es ist aber ein robusterer Ansatz, da für eine gute Schätzung nicht die gesamte Historie (t,u,v) im Trainingskorpus vorkommen muß, sondern nur eines ihrer drei möglichen Wortpaare (t,u), $(t,\cdot,v)$ oder (u,v) und jedes dieser Wortpaare mindestens genauso häufig, in der Regel aber häufiger als die gesamte Historie gesehen worden ist.

Im Gegensatz zu den bisherigen Trigramm–Ergänzungen Varigramm, Wortphrasen und Wortklassen gibt es hier kein Auswahlverfahren zur Modellbildung: die drei Trigramm–Modelle sind einfach die Randsummen des Viergramms und werden wie das normale Trigramm geglättet. Schwieriger ist das Problem, die drei Teilmodelle zu einem Sprachmodell zusammenzubinden. Es bietet sich die bekannte Methode der linearen Interpolation der drei Teilmodelle an. Als Alternative gibt es auch den zur Zeit in der Literatur diskutierten Maximum–Entropy–Ansatz. In diesem Kapitel sollen beide Methoden miteinander verglichen und ihre Stärken und Schwächen beurteilt werden.

7.1 Zusammenbindung von Abstand–Trigramm–Modellen

Die beiden Ansätze lineare Interpolation und Maximum Entropy lassen sich, wie im Rahmen dieser Arbeit herausgefunden wurde, aus einer allgemeinen Form herleiten. In der Literatur werden beide Ansätze unabhängig voneinander betrachtet. Die gemeinsame Herleitung soll hier aber voran stehen, um die Eigenschaften und Voraussetzungen der beiden Ansätze deutlich voneinander abzuheben.

Ganz abstrakt kann die Abhängigkeit eines Wortes w von seiner Historie (t,u,v) mit einem Gewicht $\alpha(t,u,v,w)$ belegt werden. Durch Renormierung ergibt sich ein Sprachmodell $p(w|t,u,v)$:

$$p(w|t,u,v) = \frac{\alpha(t,u,v,w)}{\sum_{\tilde{w}} \alpha(t,u,v,\tilde{w})} \quad .$$

Die Maximum–Likelihood–Schätzung würde ein normales Viergramm mit relativen Häufigkeiten liefern. Das Gewicht $\alpha(t,u,v,w)$ wird für die angestrebten Modelle in seine Trigramm– und Abstand–2–Trigramm–Abhängigkeiten aufgebrochen. Der Unterschied zwischen beiden Ansätzen liegt in der Art dieser Aufspaltung. Für die lineare Interpolation wird das Gewicht $\alpha(t,u,v,w)$ additiv aufgebrochen:

$$\begin{aligned}
&\frac{\alpha(t,u,v,w)}{\sum_{\tilde{w}} \alpha(t,u,v,\tilde{w})} = \\
&= \frac{\alpha'(t,\cdot,v,w) + \alpha'(t,u,\cdot,w) + \alpha'(u,v,w)}{\sum_{\tilde{w}} \alpha'(t,\cdot,v,\tilde{w}) + \alpha'(t,u,\cdot,\tilde{w}) + \alpha'(u,v,\tilde{w})} \\
&= \frac{\mu_1 \cdot p(w|t,\cdot,v) + \mu_2 \cdot p(w|t,u,\cdot) + (1-\mu_1-\mu_2) \cdot p(w|u,v)}{\sum_{\tilde{w}} \mu_1 \cdot p(\tilde{w}|t,\cdot,v) + \mu_2 \cdot p(\tilde{w}|t,u,\cdot) + (1-\mu_1-\mu_2) \cdot p(\tilde{w}|u,v)} \\
&= \mu_1 \cdot p(w|t,\cdot,v) + \mu_2 \cdot p(w|t,u,\cdot) + (1-\mu_1-\mu_2) \cdot p(w|u,v) \qquad (7.1)
\end{aligned}$$

Durch Setzung von $\alpha'(u,v,w) = (1-\mu_1-\mu_2) \cdot p(w|u,v)$ wird, analog auch für $\alpha'(t,u,\cdot,w)$ and $\alpha'(t,\cdot,v,w)$, vorausgesetzt daß

$$\begin{aligned}
\alpha'(u,v,w) > 0 \quad , \\
\sum_w \alpha'(u,v,w) = (1-\mu_1-\mu_2) \quad ,
\end{aligned}$$

und schließlich

$$\sum_w \alpha'(t,u,\cdot,w) + \alpha'(t,\cdot,v,w) + \alpha'(u,v,w) = 1 \quad .$$

Dies sind sehr strenge Randbedingungen. Für den Maximum–Entropy–Ansatz wird das Gewicht $\alpha(t,u,v,w)$ multiplikativ aufgebrochen:

$$\begin{aligned}
&\frac{\alpha(t,u,v,w)}{\sum_{\tilde{w}} \alpha(t,u,v,\tilde{w})} = \\
&= \frac{\alpha''(t,\cdot,v,w) \cdot \alpha''(t,u,\cdot,w) \cdot \alpha''(u,v,w)}{\sum_{\tilde{w}} \alpha''(t,\cdot,v,\tilde{w}) \cdot \alpha''(t,u,\cdot,\tilde{w}) \cdot \alpha''(u,v,\tilde{w})} \\
&= \frac{\exp\left[\log \alpha''(t,\cdot,v,w) + \log \alpha''(t,u,\cdot,w) + \log \alpha''(u,v,w)\right]}{\sum_{\tilde{w}} \exp\left[\log \alpha''(t,\cdot,v,\tilde{w}) + \log \alpha''(t,u,\cdot,\tilde{w}) + \log \alpha''(u,v,\tilde{w})\right]} \\
&= \frac{e^{\lambda_{t\cdot vw} + \lambda_{tu\cdot w} + \lambda_{uvw}}}{\sum_{\tilde{w}} e^{\lambda_{t\cdot v\tilde{w}} + \lambda_{tu\cdot \tilde{w}} + \lambda_{uv\tilde{w}}}} \qquad (7.2)
\end{aligned}$$

mit den Gewichten

$$\lambda_{t\cdot vw} := \log \alpha''(t,\cdot,v,w)$$

und analogen Definitionen für $\lambda_{tu\cdot w}$ und λ_{uvw}. Wie für die lineare Interpolation Gl. (7.1) muß

$$\alpha''(t,\cdot,v,w),\ \alpha''(t,u,\cdot,w),\ \alpha''(u,v,w) > 0$$

gelten. Durch die explizite Renormierung in Gl. (7.2) entfallen aber die weiteren Einschränkungen an die Gewichte. Im Folgenden wird auf die beiden Ansätze einzeln eingegangen.

7.1.1 Zusammenbindung durch lineare Interpolation

Das Prinzip der linearen Interpolation ist bereits von den Glättungsverfahren in Kapitel 3.2.1 und den Wortklassen in Kapitel 6.3 bekannt. Die Interpolationsparameter $\mu_1,\ \mu_2,\ \mu_3 := 1 - \mu_1 - \mu_2$ aus Gl. (7.1) lassen sich durch eine aus dem EM–Algorithmus [Baum 72, Dempster et al. 77, Ney 94] abgeleitete Iterationsformel schätzen:

$$\tilde{\mu}_1 = \frac{1}{N} \cdot \sum_n \frac{\mu_1 \cdot p(w_n|w_{n-3},\cdot,w_{n-1})}{\mu_1 \cdot p(w_n|w_{n-3},\cdot,w_{n-1}) + \mu_2 \cdot p(w_n|w_{n-3},w_{n-2},\cdot) + \mu_3 \cdot p(w_n|w_{n-2},w_{n-1})} \tag{7.3}$$

mit $\tilde{\mu}_1$ als neue, aus $\mu_1,\ \mu_2,\ \mu_3$ bestimmte Schätzung. Die Formeln für $\tilde{\mu}_2$ und $\tilde{\mu}_3$ lauten analog.

Eine ähnliche Schätzformel gibt es auch für die Wahrscheinlichkeitsverteilungen $p(w|t,\cdot,v),\ p(w|t,u,\cdot),\ p(w|u,v)$. Die Anwendung dieser Schätzformel hat aber immer eine Überanpassung auf dem Trainingskorpus mit schlechten Ergebnissen auf dem Testkorpus zur Folge. Eine geeignete robustere Schätzung ist bisher nicht bekannt. Für die Wahrscheinlichkeitsverteilungen wurden daher die wie üblich mit Absolute Discounting mit Interpolation und Singleton–Glättungsverteilung geglätteten relativen Häufigkeiten verwendet. Durch die Glättung werden implizit auch die Abstand–2–Bigramme $(u,\cdot,w)$, die Bigramme (v,w) und die Unigramme (w) mitmodelliert.

Als Erweiterung des interpolierten Modells Gl. (7.1) lassen sich historienabhängige Interpolationsparameter $\mu_1(h),\ \mu_2(h),\ \mu_3(h)$ einführen. Um für einen solchen historienabhängigen Interpolationsparameter noch genügend Ereignisse im Training zu haben, wird die Historienabhängigkeit lediglich an der Häufigkeit des Vorgängerwortes v zum Wort w festgemacht. Damit ergibt sich das Modell

$$p(w|t,u,v) = \mu_1(N(v)) \cdot p(w|t,\cdot,v) + \mu_2(N(v)) \cdot p(w|t,u,\cdot) + \mu_3(N(v)) \cdot p(w|u,v) \quad . \tag{7.4}$$

Für $\tilde{\mu}_1(r)$ mit r als im Training vorkommende Unigrammhäufigkeit ändert sich die iterative Schätzformel Gl. (7.3) zu

$$\tilde{\mu}_1(r) = \frac{1}{r \cdot n_r} \cdot \sum_{n:\ N(w_{n-1})=r} \frac{\mu_1(r) \cdot p(w_n|w_{n-3},\cdot,w_{n-1})}{\mu_1(r) \cdot p(w_n|w_{n-3},\cdot,w_{n-1}) + \mu_2(r) \cdot p(w_n|w_{n-3},w_{n-2},\cdot) + \mu_3(r) \cdot p(w_n|w_{n-2},w_{n-1})}$$

mit

$$n_r := \sum_{w:\ N(w)=r} 1 \quad .$$

Für $\tilde{\mu}_2(r)$ und $\tilde{\mu}_3(r)$ sind die iterativen Schätzformeln analog.

7.1.2 Zusammenbindung durch Maximum Entropy

Das Maximum–Entropy–Prinzip [Della Pietra et al. 95, Berger et al. 96] ist eine wohldefinierte Methode zur Einbindung unterschiedlicher statistischer Abhängigkeiten („Features“) i in ein Gesamtmodell und wird seit einiger Zeit auch in der statistischen Sprachmodellierung benutzt [Rosenfeld 94, Stolcke et al. 97]. Für ein Wort w gegeben seine Historie h lautet die allgemeine funktionale Form [Bishop et al. 75, pp. 83-87]:

$$p_\Lambda(w|h) = \frac{\exp\left[\sum_i \lambda_i f_i(h,w)\right]}{Z_\Lambda(h)} \tag{7.5}$$

$$Z_\Lambda(h) := \sum_{\tilde{w}} \exp\left[\sum_i \lambda_i f_i(h,\tilde{w})\right] \quad .$$

Für jedes Feature i gibt es eine Feature–Funktion

$$f_i(h,w) = \begin{cases} 1 & \text{if } i \in (h,w) \\ 0 & \text{otherwise} \end{cases}$$

und ein Gewicht $\lambda_i \in \Lambda$. Um von der allgemeinen Form in Gl. (7.5) auf die speziellere für Trigramme und Abstand–Trigramme hergeleitete Form Gl. (7.2) zu kommen, wird für jedes einzelne zu modellierende, also in der Regel für jedes im Trainingskorpus beobachtete, Trigramm (u,v,w) ein separates Feature gebildet mit der Feature–Funktion

$$f_{uvw}(\tilde{u},\tilde{v},\tilde{w}) = \begin{cases} 1 & \text{if } w=\tilde{w} \text{ and } v=\tilde{v} \text{ and } u=\tilde{u} \\ 0 & \text{otherwise} \end{cases}$$

und dem Gewicht λ_{uvw}. Gleiches gilt für die Abstand–Trigramme $(t,\cdot,v,w)$ und $(t,u,\cdot,w)$.

Die Ableitung der log–Likelihood von Gl. (7.5) nach λ_i zum Zweck der Parameterbestimmung führt auf die sogenannten „Constraint–Gleichungen“

$$\begin{aligned} Q_i(\Lambda) &= N_i \\ Q_i(\Lambda) &:= \sum_{hw} N(h)\, p_\Lambda(w|h)\, f_i(h,w) \\ N_i &:= \sum_{hw} N(h,w)\, f_i(h,w) \quad . \end{aligned} \tag{7.6}$$

Dabei läßt sich N_i als Häufigkeit des Feature i im Trainingskorpus und, mit der Näherung $p_\Lambda(h)\cdot N = N(h)$, $Q_i(\Lambda)$ als Erwartungswert der Häufigkeit des Feature i bezüglich des Maximum–Entropy–Parametersatzes Λ interpretieren. Eine allgemeine geschlossene Lösung für Gl. (7.6) ist nicht bekannt. Für den Spezialfall hierarchischer Features (Trigramm, Bigramm) ist im Rahmen dieser Arbeit eine geschlossene Lösung angenähert und im Anhang I beschrieben worden. Eine geschlossene Lösung für die Parameter des Modells Gl. (7.2) ist nicht bekannt. Sie müssen daher mittels „Generalized Iterative Scaling (GIS)“ [Darroch & Ratcliff 72] iterativ geschätzt werden. Dabei wird für diese Arbeit die beschleunigte Berechnung der Renormierung $Z_\Lambda(h)$ nach [Stolcke et al. 97]

und eine für diese Arbeit entwickelte beschleunigte Berechnung des Erwartungswertes $Q_i(\Lambda)$, beschrieben in Anhang J, verwendet.

Aufgrund sowohl der Glättung als auch der Vergleichbarkeit mit der linearen Interpolation Gl. (7.1) wurde das Modell Gl. (7.2) noch um Bigramme, Abstand–2–Bigramme und Unigramme zu

$$p_\Lambda(w|t,u,v) = \frac{e^{\lambda_{tu\cdot w}+\lambda_{u\cdot w}+\lambda_{t\cdot vw}+\lambda_{uvw}+\lambda_{vw}+\lambda_w}}{Z_\Lambda(t,u,v)} \tag{7.7}$$

ergänzt. Wahrscheinlichkeitsmasse zur Glättung wurde dadurch gewonnen, daß ähnlich dem Absolute Discounting der absoluten Häufigkeit N_i eines Features i ein Discountwert $0 < d < 1$ abgezogen wird, mit einem unterschiedlichen Discountwert für jede der in Gl. (7.7) vorkommenden sechs Feature–Arten. Da für die Constraint–Gleichungen zu Modell Gl. (7.7) keine geschlossene Lösung gefunden werden konnte, können auch keine Schätzformeln für die Discountwerte angegeben werden. Durch die Einführung der Glättung beschreiben die geänderten Constraint–Gl. (7.6) nicht mehr das Optimum der log–Likelihood des Trainingskorpus. Damit ist die Konvergenz des GIS nicht mehr garantiert. In der Praxis haben sich aber daraus keine Probleme ergeben.

7.2 Experimentelle Ergebnisse

7.2.1 Lineare Interpolation

Für die lineare Interpolation Gl. (7.1) auf dem WSJ0–Korpus wurden die Interpolationsparameter durch vereinfachte Kreuzvalidierung gebildet: Über dem Retained–Part wurde die Trigramm– und Abstand–2–Trigramm–Statistik gebildet und die Interpolationsparameter auf dem Held–Out–Part geschätzt. Dasselbe gilt im Prinzip auch für die historienabhängige lineare Interpolation Gl. (7.4). Die Zuordnung der Parameter $\mu_i(r)$ bezieht sich dabei auf die Häufigkeit $r = N(v)$ des Vorgängerwortes im gesamten Trainingskorpus. Da sich die $\mu_i(0)$ der im Training ungesehenen Vorgängerwörter nicht bestimmen lassen, werden sie zu $\mu_i(0) = \mu_i(1)$ gesetzt.

Die Ergebnisse finden sich in Tab. 7.1. Es ergeben sich Verminderungen von 8% bis 10% in der Perplexität im Vergleich zum normalen Trigramm–Sprachmodell, wobei die relative Verminderung mit der Korpusgröße zunimmt. Die historienabhängige Interpolation reduziert die Perplexität nochmals um 1–2%. Dies ist allerdings nur ein kleiner Gewinn im Vergleich zum durch die Historienabhängigkeit verursachten Aufwandes.

Für das NAB–Korpus wurde aus Speicherplatzgründen das bisher verwendete Trigramm–Sprachmodell mit den beiden kompakten Abstand–2–Trigramm–Modellen verwendet. Bei einem kompakten Modell werden die Singleton–Ereignisse $N(h,w) = 1$ weggelassen und ihre Wahrscheinlichkeitsmasse zur Glättung verwendet. Dies bewirkt eine deutliche Verminderung im Speicherplatzbedarf bei nur unmerklich verschlechterter Performanz. Zur Bestimmung der Interpolationsparameter wurden diejenigen μ_1, μ_2 und μ_3 mit guten Perplexitätsergebnissen auf dem NAB–DEV–Korpus vorab ausgewählt und

Tabelle 7.1: Testperplexitäten für Trigramme und Abstand–2–Trigramme, lineare Interpolation, WSJ0.

Modell	1M	4M	39M
Trigramm	229.7	152.1	96.8
+ Abstand–2–Trigramme	211.5	138.6	87.6
+ Historienabhängigkeit	208.2	136.3	86.5

mit diesen das Rescoring durchgeführt. Die Ergebnisse finden sich in Tabelle 7.2. Das in der Wortfehlerrate beste Modell vermindert die Perplexität um 10% und die Wortfehlerrate um 3% relativ. Dieser Gewinn bleibt auch in etwa erhalten, wenn man mit diesem besten Parametersatz auf die ungesehenen Testdaten des NAB–EVL–Korpus geht, siehe Tabelle 7.3. Nach derselben Vorgehensweise wurde auch versucht, das phrasenbasierte Trigramm–Modell mit interpolierten Wortklassen als das bisher beste Sprachmodell durch hinzuinterpolierte Abstand–Trigramme weiter zu optimieren. Dabei ist μ_3 das Gewicht des Klassenmodells. Auf dem NAB–DEV–Korpus läßt es sich auch tatsächlich um 7% in der Perplexität und um 3% relativ in der Wortfehlerrate verbessern (Tabelle 7.4). Auf dem NAB–EVL–Korpus bleiben von diesen Verminderungen noch 7% in der Perplexität und 2% relativ in der Wortfehlerrate übrig (Tabelle 7.5). Hier wurden keine Experimente mit historienabhängigen Interpolationsparametern durchgeführt.

Das Training für Verbmobil wurde auf dieselbe Art mit vereinfachter Kreuzvalidierung durchgeführt wie für WSJ0. Daraus ergibt sich eine Verminderung in der Perplexität von 8% auf wort– und phrasenbasiertem Korpus. Die Wortfehlerrate vermindert sich um 5% auf dem wort– und um 3% auf dem phrasenbasierten Korpus. Durch die Historienabhängigkeit ergibt sich nur eine geringe Verbesserung in der Perplexität. Die Wortfehlerrate ändert sich auf dem wortbasierten Korpus nicht, geht aber auf dem phrasenbasierten Korpus nochmals um 2% zurück. Wird zum linear interpolierten Sprachmodell Gl. (7.1) zusätzlich das klassenbasierte Sprachmodell Gl. (6.2) interpoliert, ergeben sich Verminderungen in der Perplexität um 3% bis 4% und in der Wortfehlerrate ebenfalls um 3% bis 4%. Dies sind ungefähr dieselben relativen Verminderungen, wie sie auch in Tabelle 7.6 für interpoliertes wortbasiertes und wortklassenbasiertes Trigramm–Sprachmodell ohne Abstand–Trigramme beobachtet wurden. Abstand–Trigramme und Wortklassen lassen sich also ohne Verluste der Einzelverbesserung zusammen verwenden. Interessanterweise hat das durch Trigramme und Abstand–2–Trigramme angenäherte Viergramm–Modell sowohl in der Wortfehlerrate als auch in der Perplexität eine bessere Performanz als das tatsächliche Viergramm–Modell. Die Ursache dafür liegt darin, daß sich auf dem kleinen Verbmobil–Korpus die Viergramm–Wahrscheinlichkeiten nicht so robust schätzen lassen wie die (Abstand–)Trigramm–Wahrscheinlichkeiten.

Tabelle 7.2: Testperplexität und Wortfehlerrate für Trigramm– und Abstand–Trigramm–Modelle, lineare Interpolation, NAB–DEV.

Modell	μ_1, μ_2	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Worttrigramm	—	21	121.8	1.8/2.0	12.8
Worttrigramm	0.4, 0.1	22	110.0	2.0/1.9	12.6
+ Abstand–Trigramme	0.3, 0.1	23	109.2	2.0/1.9	12.5
	0.2, 0.1	23	109.5	1.9/1.8	12.4
	0.1, 0.1	24	111.3	1.9/1.8	12.4

Tabelle 7.3: Testperplexität und Wortfehlerrate für Trigramm– und Abstand–Trigramm–Modelle, lineare Interpolation, NAB–EVL, Interpolationsparameter und Skalierungsfaktor optimiert auf dem NAB–DEV–Korpus.

Modell	μ_1, μ_2	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Worttrigramm	—	19	123.4	2.0/2.3	13.6
Worttrigramm					
+ Abstand–Trigramme	0.2, 0.1	23	110.3	2.2/2.1	13.2

Tabelle 7.4: Testperplexität und Wortfehlerrate für Trigramm– und Abstand–Trigramm–Modelle, lineare Interpolation, NAB–DEV, 200 Wortphrasen, 2000 Wortklassen.

Modell	μ_1, μ_2, μ_3	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Phrasen-/Klassentrigramm	—	21	114.2	1.6/1.9	12.2
+ Abstand–Trigramme	0.2, 0.1, 0.1	21	105.9	1.8/1.8	11.8

7.2.2 Maximum–Entropy

Generalized Iterative Scaling (GIS) ist ein sehr rechenintensives und daher lange laufendes iteratives Verfahren zur Bestimmung der Parameter des Maximum–Entropy–Sprachmodells Gl. (7.7), siehe dazu auch Anhang J. Aufgrund der hohen Laufzeiten konnten nur Ergebnisse auf dem kleineren Korpus WSJ0–4M ermittelt werden. Das Maximum–Entropy–Modell wird zuerst nur für die hierarchischen Features (Trigramm, Bigramm, Unigramm) für 20 Iterationen GIS–trainiert und die Abstand–n–Gramm–Features anschließend zugefügt. Es wird so lange weitertrainiert, bis auf dem Testkorpus die optimale Perplexität erreicht ist.

Tabelle 7.5: Testperplexität und Wortfehlerrate für Trigramm– und Abstand–Trigramm–Modelle, lineare Interpolation, NAB–EVL, 200 Wortphrasen, 2000 Wortklassen, Interpolationsparameter und Skalierungsfaktor optimiert auf dem NAB–DEV–Korpus.

Modell	μ_1, μ_2, μ_3	LMSc	PP	Wortfehler [%]	
				del/ins	WER
Phrasen-/Klassentrigramm	—	21	112.9	2.0/2.2	13.3
+ Abstand–Trigramme	0.2, 0.1, 0.1	21	104.5	2.0/2.2	13.0

Tabelle 7.6: Testperplexitäten und Wortfehlerraten für wortbasiertes und wortklassenbasiertes Trigramm–Sprachmodell, lineare Interpolation, Verbmobil.

Trainingstext	Modell	LMSc	PP	Wortfehler [%]	
				del/ins	WER
wortbasiert	Worttrigramm	14	40.6	3.3/3.1	17.7
	+ Abstand–2–Trigramme	18	37.5	3.7/2.6	16.9
	+ Historienabhängigkeit	17	36.7	3.5/2.7	16.9
	+ Klassentrigramm	17	36.2	3.5/2.5	16.4
	Wortviergramm	17	39.3	3.6/2.7	17.3
phrasenbasiert	Worttrigramm	14	39.5	3.4/3.0	17.2
	+ Abstand–2–Trigramme	17	36.3	3.4/2.8	16.7
	+ Historienabhängigkeit	16	35.7	3.5/2.8	16.3
	+ Klassentrigramm	17	34.9	3.5/2.7	16.1

Der Vergleich zwischen dem Maximum–Entropy–Modell Gl. (7.7) und dem interpolierten Modell mit geglätteten relativen Häufigkeiten in Tabelle 7.7 fällt deutlich zuungunsten von Maximum–Entropy aus, obwohl die Parameter für Absolute Discounting sowie die Anzahl der Iterationen auf der Testperplexität optimiert wurden und das Training somit Maximum–Entropy etwas bevorteilt. Um dieses schlechte Ergebnis genauer zu untersuchen, wurde das Testkorpus in Tabelle 7.8 genauer untersucht und nach zwei Kriterien aufgeteilt:

- Kriterium 1: Korpuspositionen, an denen alle sechs Maximum–Entropy–Features aktiv sind;
- Kriterium 2: Korpuspositionen, an denen mindestens eines der drei Trigramm– und Abstand–2–Trigramm–Features aktiv ist.

Es zeigt sich, daß an den Positionen, an denen alle sechs Features gegeben sind, Maximum–Entropy in der Perplexität um 17% unter der linearen Interpolation liegt, also deutlich überlegen ist. An den Stellen, an denen nur eines der drei Trigramm–Features gegeben ist, liegen beide Verfahren gleichauf. Ist keines der drei Trigramm–

Tabelle 7.7: Testperplexitäten für Trigramm– und Abstand–2–Trigramm–Sprachmodelle, WSJ0–4M.

Modell	PP
Trigramm, geglättete relative Häufigkeiten	152.1
Trigramm interpoliert mit Abstand–2–Trigrammen, geglättete relative Häufigkeiten	138.6
Trigramm und Abstand–2–Trigramm, Maximum–Entropy ($d_D = 0.5$, 20+3 Iter.)	146.2

Tabelle 7.8: Größe und Perplexität der Teilkorpora nach Aufteilung des WSJ0–Testkorpus nach zwei unterschiedlichen Kriterien (siehe Text), für Trigramme und Abstand–2–Trigramme, WSJ0–4M.

Kriterium	Positionen	PP	
		Maximum–Entropy	Interpolation
1: wahr	83 208	7.7	9.3
1: falsch	241 447	404.2	351.4
2: wahr	193 278	27.5	27.5
2: falsch	131 377	1713.4	1499.4
alle	324 655	146.2	138.6

Features gegeben, so liegt Maximum–Entropy in der Perplexität um 14% über der linearen Interpolation, ist also deutlich schlechter. Die Vermutung ist, daß die Glättung für Maximum–Entropy–Modelle mit nicht–hierarchischen Features noch nicht optimal durchgeführt wird.

Um diese Vermutung weiter zu untermauern, wurde für die ungeglätteten Modelle die Perplexität auf dem Trainingskorpus berechnet. Dabei wurden die ersten beiden Wörter eines Satzes nicht berücksichtigt, da es dafür keine Abstand–2–Trigramme gibt. Hier war es nun auch möglich, die interpolierten Wahrscheinlichkeiten und nicht nur die Interpolationsparameter durch den EM–Algorithmus zu bestimmen. Wie aus Tabelle 7.9 zu ersehen, ist aber trotzdem das Maximum–Entropy–Modell deutlich besser als das Interpolationsmodell. Also scheint Maximum–Entropy tatsächlich besser als die Interpolation zu sein, wenn keine Glättung nötig ist.

Es wurde deshalb eine vereinfachte Aufgabe definiert, bei der die Features wesentlich besser trainiert sind als für das Modell Gl. (7.7). Dabei handelt es sich um die Erweiterung des Bigramms um das Abstand–2–Bigramm, also das linear interpolierte Modell

$$p(w|u,v) = \mu \cdot p(w|u,\cdot) + (1-\mu) \cdot p(w|v)$$

Tabelle 7.9: Perplexitäten auf dem WSJ0–4M Trainingskorpus für ungeglättete Maximum–Entropy– und Interpolationsmodelle basierend auf Trigrammen und Abstand–2–Trigrammen.

Modell	PP
Interpolation:	
relative Häufigkeiten als Wahrscheinlichkeiten	5.8
EM–trainierte Wahrscheinlichkeiten (20 Iter.)	4.8
Maximum–Entropy (20 Iter.)	2.6

Tabelle 7.10: Testperplexitäten für Bigramm und Abstand–2–Bigramm, WSJ0–4M.

Modell	PP
Bigramm, geglättete relative Häufigkeiten	205.4
Bigramm interpoliert mit Abstand–2–Bigramm, geglättete relative Häufigkeiten	193.4
Bigramm und Abstand–2–Bigramm, Maximum–Entropy ($d_D = 0.6$, 20+5 Iter.)	166.9

mit separat geglätteten Verteilungen $p(w|u,\cdot)$ und $p(w|v)$, die damit auch das Unigramm implizieren, sowie das analoge Maximum–Entropy–Modell

$$p_\Lambda(w|u,v) = \frac{e^{\lambda_{u\cdot w}+\lambda_{vw}+\lambda_w}}{Z_\Lambda(u,v)} \quad .$$

Das Training dieser Modelle wurde ganz analog dem der Abstand–2–Trigramm–Modelle durchgeführt. Die Ergebnisse in Tabelle 7.10 zeigen tatsächlich einen Verminderung in der Testperplexität um 6% für die lineare Interpolation, jedoch um 19% für die Maximum–Entropy.

Zusammenfassung. Die Hinzunahme von Abstand–2–Trigrammen zum bisherigen Trigramm mittels linearer Interpolation ist unproblematisch und erbringt eine Reduktion in der Perplexität um bis zu 10% und in der Wortfehlerrate um bis zu 5% relativ. Historienabhängige Interpolationsparameter sind etwas aufwendiger und bringen nur noch geringe Verbesserungen. Die Hinzunahme von Wortphrasen und Wortklassen reduziert die jeweiligen Verminderungen in Perplexität und Wortfehlerrate kaum, so daß sich alle drei Ansätze gut miteinander kombinieren lassen. Maximum–Entropy als Alternative zur linearen Interpolation kann sich bei gut trainierten Features durchsetzen. Bei selteneren Features wie den Abstand–2–Trigrammen ist aber die Glättung bisher noch nicht gut genug konzipiert, um an die Performanz der linearen Interpolation heranzukommen oder diese gar zu übertreffen.

Kapitel 8

Zusammenfassung der Erkenntnisse

In Kapitel 8 werden die Ergebnisse aus den vorangegangenen Kapiteln zusammengefasst und es wird eine Beantwortung der Problemstellung aus Kapitel 2.3 versucht. Im Vordergrund stehen dabei die Änderungen in der Wortfehlerrate auf dem Verbmobil– und dem NAB–Korpus. Aus Platzgründen wird das WSJ0–Korpus nur noch in Ausnahmefällen erwähnt.

1. Glättungsverfahren

Die Performanz der verschiedenen Glättungsmethoden wird nochmals in den Tabellen 8.1 für das Verbmobil–Korpus und 8.2 für das NAB–Korpus dargestellt. Es lassen sich folgende allgemeine Aussagen treffen:

- Absolute Discounting ist besser als Linear Discounting (Perplexität 11–16%, Wortfehlerrate 4–9% relativ). Der relative Gewinn ist auf allen beiden Korpora etwa gleich groß und etwa ebenso groß wie in [Ney et al. 94] angegeben.
- Absolute Discounting vorausgesetzt, läßt sich zwischen Backing–Off und Interpolation keine klare Präferenz treffen. Auf dem Verbmobil–Korpus ist Interpolation besser als Backing–Off (Perplexität 4%, Wortfehlerrate 5% relativ), auf NAB verhält es sich dagegen umgekehrt (Perplexität 5–6%, Wortfehlerrate 1–2% relativ). Eine genaue Untersuchung der Ursache ist aufgrund des folgenden Ergebnisses nicht durchgeführt worden.
- Die Singleton–Glättung erbringt nämlich eine weitere Verbesserung (Perplexität 3% auf Verbmobil, 8–9% auf NAB, Wortfehlerrate 1% relativ auf Verbmobil, 1–4% relativ auf NAB), und zwar diesmal auf allen Korpora mit der Methode der Interpolation. Diese Verbesserung ist für Verbmobil deutlich und für NAB etwas geringer ausgefallen als in [Kneser & Ney 95] angegeben, kann aber für Verbmobil durch das vergleichsweise kleine Korpus erklärt werden.

Das durchgängig, auch auf den anderen Korpora, optimale Glättungsverfahren ist somit das absolute Discounting mit Interpolation und der Singleton–Glättungsvertei-

Tabelle 8.1: Testperplexitäten und Wortfehlerraten für Glättungsverfahren, Verbmobil.

Glättungsverfahren	PP	Wortfehler [%]	
		del/ins	WER
Linear Discounting:			
Backing–Off	52.4	3.9/3.4	19.6
Interpolation	48.2	3.3/4.1	18.9
Absolute Discounting:			
Backing–Off	43.8	3.5/3.2	18.8
Interpolation	41.9	3.3/3.2	17.9
Absolute Discounting, Singleton–Glättung:			
Backing–Off	43.9	3.6/3.0	18.3
Interpolation	40.6	3.3/3.1	17.7

Tabelle 8.2: Testperplexitäten und Wortfehlerraten für Glättungsverfahren, NAB.

Glättungsverfahren	DEV			EVL		
	PP	Wortfehler [%]		PP	Wortfehler [%]	
		del/ins	WER		del/ins	WER
Linear Discounting:						
Backing–Off	144.7	1.7/2.6	13.9	150.4	1.8/3.0	14.8
Interpolation	148.8	1.5/3.1	14.1	156.5	1.5/3.4	14.8
Absolute Discounting:						
Backing–Off	125.8	1.7/2.4	13.3	127.6	1.9/2.5	13.5
Interpolation	132.0	1.5/2.7	13.4	135.4	1.6/2.9	13.8
Absolute Discounting, Singleton–Glättung:						
Backing–Off	122.9	1.8/2.0	13.1	123.7	2.1/2.3	13.6
Interpolation	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6

Tabelle 8.3: Testperplexitäten und Wortfehlerraten für Varigramm–Modelle, Verbmobil, 500 gewählte Historien.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Trigramm	40.6	3.3/3.1	17.7
Varigramm	39.5	3.7/2.7	17.5

Tabelle 8.4: Testperplexitäten und Wortfehlerraten für Varigramm–Modelle, NAB, 10 000 gewählte Historien.

Modell	DEV			EVL		
	PP	Wortfehler [%]		PP	Wortfehler [%]	
		del/ins	WER		del/ins	WER
Trigramm	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6
Varigramm	117.9	1.6/2.1	12.7	119.2	2.0/2.3	13.4

lung. Gute Ergebnisse wurden auch in [Seymore et al. 97] für ein ähnliches Glättungsverfahren beobachtet. Absolute Discounting mit Interpolation ist für alle weiteren Versuche verwendet worden. Im Vergleich zum Linear Discounting mit Interpolation als ursprünglichem Glättungsverfahren ergibt sich auf dem Verbmobil–Korpus eine Verbesserung von 16% in der Perplexität und 6% relativ in der Wortfehlerrate für Verbmobil und um 18–21% in der Perplexität und 8–9% relativ in der Wortfehlerrate für NAB. Interessanterweise sind die relativen Gewinne für das große Korpus am deutlichsten, obwohl dieses von vorneherein besser geschätzt wird. Dies unterstreicht, daß die Glättungsverfahren auch auf einer Statistik beruhen, die mit zunehmender Korpusgröße immer besser bestimmt wird.

2. Varigramme

Das besten Ergebnisse für Varigramme sind in Tabelle 8.3 und in Tabelle 8.4 dargestellt. Es ergibt sich nahezu gleichmäßig auf allen Korpora nur eine Verbesserung in der Perplexität um 3% und in der Wortfehlerrate um 1% relativ im Vergleich zum Trigramm. Damit erbringt die Erweiterung des Trigramms um ganze bedingte Verteilungen anstelle einzelner Varigramme wie in [Kneser 96] auch keine besseren Ergebnisse. Vermutlich sind die ausgewählten Verteilungen trotz des Leaving–One–Out–Auswahlkriteriums immer noch vergleichsweise selten im Training beobachtet worden und damit für ungesehene Testdaten zu speziell. Da sich diese Verteilungen an der Spitze der Glättungshierarchie befinden, wirkt sich dies besonders deutlich aus. Der Aufwand für die Viergramm–Suche steht in keinem Verhältnis zu diesen geringen Verbesserungen. Der Varigramm–Ansatz ist daher für die weitere Arbeit nicht mehr berücksichtigt worden, zumal es mit den Wortphrasen eine Alternative gibt.

Tabelle 8.5: Testperplexität und Wortfehlerrate, Verbmobil, 100 Phrasen nach hierarchischem Unigramm–Kriterium.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Worttrigramm	40.6	3.3/3.1	17.7
Phrasentrigramm	39.5	3.4/3.0	17.2

3. Wortphrasen

Für die Wortphrasen läßt sich Folgendes feststellen:

- Das tatsächliche Auswahlkriterium für Wortphrasen, Unigramm oder Bigramm, flach oder hierarchisch, hat wenig Einfluß auf die Verminderung von Perplexität und Wortfehlerrate. Ausschlaggebend scheint tatsächlich hauptsächlich die Häufigkeit einer Wortphrase zu sein. Dies deckt sich mit [Klakow 98a]. Für diese Arbeit wurden die Wortphrasen nach dem Unigramm–log–Likelihood–Kriterium ausgewählt.
- Solange sich die Anzahl der Wortphrasen im (sehr flachen) Optimum bewegt, hat es wenig Auswirkung auf die Perplexität, ob zur Bestimmung der Wahrscheinlichkeiten das korrekte Summenkriterium verwendet wird oder eine Vereinfachung davon. Für die weiteren Arbeiten wurde daher das einfachste Parsing–Kriterium, maximale Abdeckung, gewählt.
- Die besten Ergebnis für Wortphrasen sind in Tabelle 8.5 und in Tabelle 8.6 dargestellt. Es ergibt sich eine Reduktion in der Perplexität um 2–4% und in der Wortfehlerrate um 1–3% relativ. Interessanterweise haben also phrasenbasierte Sprachmodelle etwa dieselbe Reduktion in der Perplexität wie die Varigramme, jedoch eine etwas bessere Performanz in der Fehlerrate. Die Literatur gibt ähnliche Verminderungen an mit Ausnahme der deutlich besseren Ergebisse in [Klakow et al. 98], deren Ursache aber zum Teil in dem erneuten akustischen Training mit Wortphrasen zu suchen ist.

Wegen der akzeptablen Reduktion in der Fehlerrate und des verhältnismäßig geringen Aufwandes sind die Wortphrasen geeignet für den weiteren Einsatz in der Sprachmodellierung und die bessere Alternative zu den Varigrammen.

4. Wortklassen

Der Austauschalgorithmus läßt sich durch eine laufzeiteffiziente Implementierung sowohl für das Bigramm– als auch für das Trigramm–Kriterium auch auf großen Korpora verwenden. Die Laufzeiten sind für das WSJ0–Korpus in Tabelle 8.7 angegeben. Die resultierenden Wortklassen lassen syntaktische und semantische Zusammenhänge erkennen, haben aber noch einige unpassende zugeordnete Wörter.

Aus Tabelle 8.8 ist jedoch zu ersehen, daß das Trigramm–Kriterium aufgrund einer Überanpassung nur auf großen Korpora zu besseren Testperplexitäten als das

Tabelle 8.6: Testperplexität und Wortfehlerrate, NAB, 200 Phrasen nach flachem Unigramm–Kriterium.

Modell	DEV			EVL		
	PP	Wortfehler [%]		PP	Wortfehler [%]	
		del/ins	WER		del/ins	WER
Worttrigramm	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6
Phrasentrigramm	119.1	1.7/2.0	12.6	118.0	1.9/2.2	13.4

Tabelle 8.7: Laufzeit pro Iteration für den Austausch–Algorithmus, WSJ0–39M, auf einer R4400 SGI Workstation, in Abhängigkeit von der Klassenanzahl G.

Kriterium	G	Laufzeit
Bigramm	200	ca. 7 min
	2000	ca. 3 h
Trigramm	200	ca. 6 h

Tabelle 8.8: Testperplexitäten für Wort– und Klassentrigramm–Modelle, WSJ0.

Modell	G	1M	4M	39M
Worttrigramm	—	229.7	152.1	96.8
Klassentrigramm	200	479.9	258.9	206.2
Klassentrigramm	200	321.9	245.8	218.5
(Bigramm–Kriterium)	2000	263.1	164.2	113.8

Bigramm–Kriterium führt, die aber auch nur gering sind. Für die weiteren Arbeiten wurde daher nur mit dem Bigramm–Kriterium gearbeitet. In jedem Fall ist das wortklassenbasierte Sprachmodell alleine deutlich schlechter als das wortbasierte.

Auch auf Verbmobil, wie aus Tabelle 8.9 zu sehen, ist das wortklassenbasierte Sprachmodell in der Perplexität schlechter als das wortbasierte, in der Fehlerrate jedoch gleichwertig. Eine Verbesserung in der Testperplexität um 7% und in der Wortfehlerrate um 2% relativ ergibt sich erst durch die Interpolation der beiden Modelle. Auf den NAB–Korpora, Tabelle 8.10, ist das reine wortklassenbasierte Trigramm–Sprachmodell etwas schlechter als das wortbasierte. Die Interpolation beider Modelle ergibt eine Verminderung in der Perplexität von 4% und in der Wortfehlerrate von 2–3%.

Die Perplexitäten liegen im Bereich dessen, was aus der Literatur zu erwarten ist, allein die guten Ergebnisse von [Kneser & Ney 93] konnten nicht wiederholt werden. Da es auch sonst keine Literaturquelle gibt, in der die klassenbasierten Sprachmodelle alleine das wortbasierte Sprachmodell schlagen, liegen diese guten Ergebnisse offen-

Tabelle 8.9: Testperplexitäten und Wortfehlerraten für Worttrigramm und Wortklassenmodell, Verbmobil, Interpolationsparameter kreuzvalidiert.

Modell	G	PP	Wortfehler [%]	
			del/ins	WER
Worttrigramm	—	40.6	3.3/3.1	17.7
+ Klassentrigramm	100	37.9	3.7/2.7	17.3
Klassentrigramm	500	41.8	3.6/3.0	17.7

Tabelle 8.10: Testperplexitäten und Wortfehlerraten für Worttrigramm und Wortklassenmodell, NAB, Interpolationsparameter auf dem NAB–DEV–Korpus optimiert.

Modell	G	DEV			EVL		
		PP	Wortfehler [%]		PP	Wortfehler [%]	
			del/ins	WER		del/ins	WER
Worttrigramm	—	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6
+ Klassentrigramm	2000	116.7	1.8/2.1	12.4	118.7	2.0/2.2	13.3
Klassentrigramm	5000	128.9	1.8/2.1	13.0	130.6	2.2/2.4	14.0

bar in der Besonderheit des dort verwendeten LOB–Korpus. Klassenbasierte Sprachmodelle interpoliert mit wortbasierten Sprachmodellen haben somit einen positiven Effekt auf Perplexität und Wortfehlerrate und sind damit eine sinnvolle Erweiterung.

5. Abstand–Trigramme

Die lineare Interpolation der beiden geglätteten Abstand–2–Trigramme mit dem Standardtrigramm erbringt, wie in Tabelle 8.11 und Tabelle 8.12 zu sehen, auf Verbmobil und NAB eine Reduktion in der Perplexität um 8–11% und in der Wortfehlerrate um 3% relativ. Ergebnisse aus der Literatur gibt es für das interpolierte Modell nicht. Die Maximum–Entropy–Methode erreicht in etwa dieselbe Verminderung der Perplexität wie in [Rosenfeld 94] angegeben, jedoch ist diese Verminderung nicht so stark wie die der linearen Interpolation geglätteter Modelle, vermutlich aufgrund eines bisher fehlenden effizienten Glättungsverfahrens für Maximum Entropy. Die interpolierten Abstand–Trigramme erbringen somit eine Verbesserung im üblichen Rahmen, allerdings um den Preis eines verdreifachten Speicheraufwandes und einer Viergrammsuche.

6. Gemeinsame Verwendung der ausgewählten Wortabhängigkeiten

Der gemeinsame Einsatz der Ergänzungen zum Standard–Worttrigramm ist in Tabelle 8.13 und Tabelle 8.14 dargestellt. Die Reduktionen in der Wortfehlerrate von ca. 3% für die einzelnen Methoden, bezogen auf das Standardtrigram, bleiben auf dem Verbmobil–Korpus auch beim gemeinsamen Einsatz der Methoden erhalten. Auf das Standardtrigramm bezogen ergibt sich so eine Gesamtreduktion von 15% in der Per-

Tabelle 8.11: Testperplexitäten und Wortfehlerraten für Worttrigramm und Abstand–Trigramme, Verbmobil, Interpolationsparameter kreuzvalidiert.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Worttrigramm	40.6	3.3/3.1	17.7
+ Abstand–Trigramm	37.5	3.7/2.6	16.9

Tabelle 8.12: Testperplexitäten und Wortfehlerraten für Worttrigramm und Abstand–Trigramme, NAB, Interpolationsparameter auf DEV optimiert.

Modell	DEV			EVL		
	PP	Wortfehler [%]		PP	Wortfehler [%]	
		del/ins	WER		del/ins	WER
Worttrigramm	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6
+ Abstand–Trigramm	109.5	1.9/1.8	12.4	110.3	2.2/2.1	13.2

plexität und um 11% relativ in der Wortfehlerrate. Bezogen auf die ursprüngliche Glättungsmethode lineares Discounting mit Interpolation ergibt sich sogar eine Reduktion um 29% in der Perplexität und um 16% relativ in der Fehlerrate. Etwas abgeschwächt trifft dies auch auf das NAB–DEV–Korpus zu. Bezogen auf das Standardtrigramm ergibt sich eine Verminderung um 13% in der Perplexität und um 8% relativ in der Wortfehlerrate. Bezogen auf die lineare Interpolation reduziert sich die Perplexität um 29% und die Wortfehlerrate um 16% relativ. Auf dem NAB–EVL–Korpus ergibt sich, bezogen auf das Standardmodell, in der Perplexität eine Verminderung um 15%, in der Wortfehlerrate jedoch nur um 4% relativ. Da die Perplexität ebenso stark sinkt wie auf dem NAB–DEV–Korpus, hat diese schwache Performanz vermutlich die Ursache in den ebenfalls nicht auf NAB–EVL optimierten akustischen Parametern. Bezogen auf die lineare Interpolation ergibt sich für das NAB–EVL–Korpus eine Reduktion in der Perplexität um 33% und in der Wortfehlerrate um 12% relativ.

Die letzte Untersuchung betrifft eine Gegenüberstellung der Performanz der Glättungsverfahren und Trigramm–Erweiterungen mit dem damit verbundenen Aufwand, wobei Auswahl und Training der Sprachmodelle als einmaliger Aufwand außer Acht gelassen wird. Es zählen nur die Laufzeiten für das Rescoring auf den Wortgraphen und der Speicherplatzbedarf der Sprachmodelle. Da der Speicherplatz für die Verbmobil–Sprachmodelle vernachlässigbar ist, werden in Tabelle 8.15 nur die Ergebnisse für das NAB–Korpus präsentiert. Der obere Teil entspricht den Einträgen in Tabelle 8.14:

- Lineare Interpolation ist zwar das schlechteste, aber auch das „billigste“ Glättungsverfahren.

Tabelle 8.13: Testperplexität und Wortfehlerrate für Glättungsverfahren und Sprachmodelle, Verbmobil.

Modell	PP	Wortfehler [%]	
		del/ins	WER
Linear Discounting (Int.)	48.2	3.3/4.1	18.9
Absolute Discounting (Int.)	41.9	3.3/3.2	17.9
+ Singleton–Glättung	40.6	3.3/3.0	17.7
+ Wortphrasen	39.5	3.4/3.0	17.2
+ Wortklassen	36.2	3.7/2.6	16.5
+ Abstand–Trigramme	34.9	3.5/2.7	16.1
+ Unigramm–Pooling für Interpolationsfaktoren	34.4	3.5/2.7	15.9

Tabelle 8.14: Testperplexität und Wortfehlerrate für Glättungsverfahren und Sprachmodelle, NAB.

Modell	DEV			EVL		
	PP	Wortfehler [%]		PP	Wortfehler [%]	
		del/ins	WER		del/ins	WER
Linear Discounting (Int.)	148.8	1.5/3.1	14.1	156.5	1.5/3.4	14.8
Absolute Discounting (Int.)	132.0	1.5/2.7	13.4	135.4	1.6/2.9	13.8
+ Singleton–Glättung	121.8	1.8/2.0	12.8	123.4	2.0/2.3	13.6
+ Wortphrasen	119.1	1.7/2.0	12.6	118.0	1.9/2.2	13.4
+ Wortklassen	114.2	1.6/1.9	12.2	112.9	2.0/2.2	13.3
+ Abstand–Trigramme	105.9	1.8/1.8	11.8	104.5	2.0/2.2	13.0

- Absolute Discounting verdreifacht die Laufzeit, jedoch bleibt der Speicherplatzbedarf natürlich konstant.
- Durch die Singleton–Glättung erhöht sich der Speicherplatzbedarf leicht um die Singleton–Statistiken, die Laufzeit bleibt jedoch gleich.
- Ebenfalls eine leichte Erhöhung des Speicherplatzbedarfes aufgrund des erweiterten Vokabulars und der damit größeren Zahl an Trigrammen ergibt sich durch die Wortphrasen. Auch dies hat kaum Auswirkungen auf die Laufzeit.
- Duch die Interpolation der Wortklassen erhöht sich der Speicherplatzbedarf durch die Statistik über die wortklassenbasierten Trigramme um 53%. Da nun zwei Modelle gleichzeitig berechnet werden, verdoppelt sich auch die Laufzeit beinahe.
- Durch die weitere Interpolation der kompakten Abstand–Trigramme verdoppelt sich der Speicherplatzbedarf. Würden volle Abstand–Trigramm–Modelle verwendet werden, würde sich der Speicherplatzbedarf sogar verdreifachen. Aufgrund der Viergramm–Suche und der vier gleichzeitig zu berechnenden Sprachmodelle vervielfacht sich die Laufzeit um den Faktor 18.

Folgende Bemerkungen sollen noch ergänzt werden:

- Der Laufzeitvergleich für das Abstand–Trigramm–Modell ist nicht ganz fair. Für die Trigramm–Sprachmodelle existiert nämlich ein Cache, der bereits berechnete Wahrscheinlichkeiten ablegt und bei Bedarf wieder ohne erneute Berechnung ausgibt. Der zweite Teil von Tabelle 8.15 gibt den Wert für Absolute Discounting mit Interpolation ohne diesen Cache an. Danach führt dieser Cache zu einer Halbierung bis Drittelung der Laufzeit. Würde einem Viergramm–Cache dieselbe Performanz unterstellt, würden sich die Laufzeiten der Abstand–Trigramm–Sprachmodelle im Vergleich zu dem wort– und wortklassenbasierten Sprachmodell „nur“ verneunfachen.
- Absolute Discounting mit Backing–Off hat zwar auf dem NAB–Korpus eine etwas bessere Performanz als Absolute Discounting mit Interpolation. Duch die Renormierung wird diese leichte Verbesserung aber mit einer knappen Verzwanzigfachung der Laufzeiten erkauft. Ohne das Caching der Trigramm–Wahrscheinlichkeiten würden die Laufzeiten sogar um den Faktor 25–28 steigen.
- Die angegebenen Laufzeiten beziehen sich nur auf das Rescoring auf einem Wortgraphen. Eine akustische Erkennung ist wesentlich zeitaufwendiger. Die Erkennung, die für die Erstellung des reinen Wortgraphen auf dem NAB–DEV–Korpus durchgeführt wurde, hatte z.B. einen Echtzeitfaktor (RTF) von ca. 50. Verwendet wurde dabei das Trigramm–Modell mit absolutem Discounting, Interpolation und Singleton–Glättung, das beim Rescoring in Tabelle 8.15 auf demselben Korpus lediglich einen Echtzeitfaktor von 0.08 hat. Für eine Erkennungsaufgabe relativieren sich somit die Laufzeitangaben aus Tabelle 8.15 erheblich.

Die durch die Sprachmodellierung erzielten Verminderungen in der Perplexität und Wortfehlerrate lassen sich ziemlich genau jeweils zur Hälfte auf die Glättungsverfahren und die Erweiterungen des Trigramm–Sprachmodells aufteilen. Die Glättungsverfahren sind kaum mit Aufwand verbunden, bei den Erweiterungen vervielfacht er sich hingegen.

Ausblick. Die kurzreichweitigen Abhängigkeiten in der Sprachmodellierung sind mit den beschriebenen Methoden umfassend ausgenutzt, so daß deutliche weitere Verbesserungen in diesem Bereich kaum zu erwarten sind. Erfolgversprechend erscheint hingegen die Betrachtung von Abhängigkeiten innerhalb eines ganzen Satzes oder Textabschnitts zu sein, die mit Maximum Entropy oder einer verwandten Methode mit den kurzreichweitigen Abhängigkeiten verknüpft werden. Solche Ansätze sind bereits in [Rosenfeld 94] für Worttrigger und in [Stolcke et al. 97] für stochastische Grammatiken beschrieben worden. Dabei sind jedoch drei Probleme zu beachten:

- Durch langreichweitigere Abhängigkeiten vergrößert sich die Worthistorie und damit der Aufwand für das Rescoring. Alternativ zum bisher verwendeten Suchverfahren müsste dann eine einfache Neubewertung der n höchstbewerteten Satzhypothesen erfolgen (sogenanntes n-best-Rescoring), die aber wegen dieser Schranke n den gesprochenen Satz möglicherweise gar nicht beinhalten. Dies betrifft insbe-

Tabelle 8.15: Speicherplatzbedarf (in MByte) und Laufzeiten (CPU–Zeit in Sekunden und Echtzeitfaktor (Real Time Factor, RTF)) für Glättungsverfahren und Sprachmodelle, NAB (* = ohne Trigram Caching; $^+$ = volle Abstand–Trigramm–Modelle).

Modell	Speicher	CPU–Zeit		RTF	
		DEV	EVL	DEV	EVL
Linear Discounting (Int.)	214	91.6	107.9	0.03	0.03
Absolute Discounting (Int.)	214	233.5	291.5	0.08	0.09
+ Singleton–Glättung	237	234.5	288.5	0.08	0.09
+ Wortphrasen	263	238.1	305.8	0.08	0.09
+ Wortklassen	403	413.7	494.2	0.15	0.15
+ Abstand–Trigramme	832	7632.5	8620.9	2.61	2.70
+ Abstand–Trigramme$^+$	1159	—	—	—	—
Absolute Discounting (BO)	214	4441.9	5273.7	1.52	1.65
Absolute Discounting* (BO)	214	14961.7	17725.2	5.12	5.55
Absolute Discounting* (Int.)	214	589.5	622.4	0.20	0.19

sondere längere Sätze, für die sich langreichweitigere Abhängigkeiten am besten eignen.

- Die Glättungsproblematik für Maximum Entropy muß noch gelöst werden. Einen ersten Ansatz in diese Richtung bildet die log–linearen Verknüpfung, die mit Absolute Discounting geglättete Teilsprachmodelle ähnlich wie Maximum Entropy einbindet und somit die Vorteile beider Ansätze vereinigt. Die ersten Ergebnisse sind durchaus positiv [Klakow 98b].

- Der Rechenaufwand steigt zum einen mit der Zahl der verwendeten Modelle und ist zum anderen durch die Verwendung von Maximum Entropy aufgrund der Renormierung ohnehin sehr hoch. Mit der stets steigenden Leistungsfähigkeit wird sich aber der erste Aspekt in den nächsten Jahren von selbst erledigen. Im Zusammenhang mit den bereits erwähnten log–linearen Modellen gibt es auch schon Ansätze, die das Problem der Renormierung umgehen. Sie sind aber bislang noch nicht veröffentlicht.

Anhang A

Textkorpora und Wortgraphen

In diesem Anhang werden die in dieser Arbeit verwendeten Korpora und Wortgraphen kurz charakterisiert.

1. Wall–Street–Journal–Korpus (WSJ0)

Das Wall–Street–Journal–Textkorpus ist Teil des „ARPA Continuous Speech Recognition Pilot Corpus“ (WSJ0) aus dem Jahr 1992. Es besteht aus Artikeln des Wall–Street–Journal der Jahre 1987–1989. Das Korpus wurde nach Vorgaben von [Rosenfeld 94] in einen Testkorpus mit 324 655 Wörtern und drei Trainingskorpora mit etwa 1, 4 und 39 Millionen Wörtern aufgeteilt, um die Effekte für unterschiedlich umfangreiche Trainingskorpora beobachten zu können. Ein kleineres Trainingskorpus ist dabei Teilmenge des nächst größeren. Das akustische Vokabular besteht aus 19 979 Wörtern. Hinzu kommen zwei Markierungen, jeweils eine für unbekannte Wörter und für das Satzende. Eine Übersicht über den Umfang der Trainingskorpora und über ihre weiteren Eigenschaften gibt Tabelle A.1.

Aus allen Korpora wurden Satzzeichen sowie Artikel–, Paragraphen– und Satzanfangsmarkierungen entfernt. Die Satzendemarkierung wurde als Satztrenner in den Texten belassen, ist Teil des Vokabulars und wird von den Sprachmodellen bewertet. Ebenso ist die Markierung, mit der diejenigen Wörter ersetzt werden, die nicht im akustischen Vokabular stehen (sogenannte Out–of–Vocabulary (OOV) Wörter), Teil des Vokabulars und wird von den Sprachmodellen bewertet. Diese Aufbereitung gilt auch für die beiden anderen Korpora. Die OOV-Rate auf dem Testkorpus beträgt 2.2%.

2. North–American–Business–Korpus (NAB)

Das North–American–Business–Korpus ist Teil der dritten „ARPA Continuous Speech Recognition Benchmark Speech Test Collection“ (CSR–III) aus dem Jahr 1994. Das akustische Material besteht aus vorgelesenen Artikeln aus den Finanzteilen von Reuters News Service, New York Times, Washington Post, Los Angeles Times und dem Wall Street Journal von April bis Juni 1994. Das Textmaterial ist eine Obermenge des WSJ0–Korpus, bestehend aus Artikeln des Wall–Street–Journal

Tabelle A.1: Verschiedene Statistiken über die WSJ0–Trainingskorpora.

Trainingskorpus		1M	4M	39M
Sätze		37 831	187 892	1 611 571
Wörter		892 333	4 472 827	38 532 518
OOV [%]		2.3	2.3	2.4
Unigramme:	total (N)	892 333	4 472 827	38 532 518
	verschiedene ($\sum_{r>0} n_r$)	17 189	19 725	19 981
	singleton (n_1)	2 465	235	0
	n_1/N	0.0028	0.0001	0.0000
Bigramme:	total (N)	892 333	4 472 827	38 532 518
	verschiedene ($\sum_{r>0} n_r$)	285 692	875 497	3 500 633
	singleton (n_1)	199 493	562 549	2 046 462
	n_1/N	0.2236	0.1258	0.0531
Trigramme:	total (N)	854 502	4 284 935	36 920 947
	verschiedene ($\sum_{r>0} n_r$)	587 985	2 370 914	14 039 536
	singleton (n_1)	510 043	1 963 267	10 897 166
	n_1/N	0.5969	0.4582	0.2951

von 1987–1994, der Agency Press von 1988–1990, und des San Jose Mercury von 1991. Die Größe der Texte finden sich in Tabelle A.2.

Das akustische Vokabular besteht aus 19 977 Wörtern zzgl. 2 434 Aussprachevarianten. Auch hier gibt es jeweils eine Markierung für unbekannte Wörter und das Satzende. Das akustische Material ist nach Männern und Frauen getrennt, es wurde jedoch geschlechtsunabhängig trainiert. Das akustische Modell wurde mit LDA und 9 375 (zzgl. Silencemodell) Triphonen modelliert, die mittels CART–Verfahren zu 3 000 + 1 tied states zusammengefasst und diese jeweils durch eine Gauß'sche Mischverteilungen dargestellt werden, wobei sich insgesamt 268 692 Einzelverteilungen ergeben. Dem Vokabular wurden im Rahmen der Arbeit 200 Wortphrasen hinzugefügt, jedoch ohne das akustische Modell neu zu trainieren.

Es wurden für Männer und Frauen separate Wortgraphen erstellt. Da in dieser Arbeit geschlechtsspezifische Aspekte nicht betrachtet wurden, sind sämtliche Ergebnisse ohne diese Trennung angegeben worden. Für die Berechnung der Wortfehlerraten gibt es zwei Korpora: Das Development–Korpus (NAB–DEV–Korpus) zur Parameteroptimierung sowie das Evaluierungskorpus (NAB–EVL–Korpus) mit ungesehenen Daten als „hartem" Testfall. Auf beiden Korpora sprechen 10 Frauen und 10 Männer insgesamt 310 (DEV) bzw. 316 (EVL) Sätze. Die Wortgraphen wurden mit integrierter Trigramm–Suche gebildet. Weitere Angaben zu den Wortgraphen finden sich in Tabelle A.3. Tabelle A.4 gibt die Graph Error Rate an. Dies ist der Fehler der Pfade im Wortgraphen mit dem kleinsten Levenshtein–Abstand zum tatsächlich gesprochenen Text. Die Graph Error Rate bildet somit die größte untere Schranke für die Wortfehlerrate auf diesem Wortgraphen.

Tabelle A.2: Größe des wort– bzw. phrasenbasierten Vokabulars, des Trainings– (Tr.), des Development– (DEV) und des Evaluierungs–Korpus (EVL) sowie der Out–of–Vocabulary–Rate (OOV) für das NAB–Korpus.

	Vokabular	Tr. (OOV[%])	DEV (OOV[%])	EVL (OOV[%])
wortbasiert	19 979	240 875 674 (2.9)	7 387 (2.7)	8 186 (2.5)
phrasenbasiert	20 179	224 125 898 (3.1)	6 854 (2.9)	7 567 (2.7)

Tabelle A.3: Word Graph Density (WGD), Node Graph Density (NGD), Bounds Graph Density (BGD) für wortbasierte und phrasenbasierte Wortgraphen des NAB–Korpus (LMScale = 17, LM– und Lattice–Pruning–Thresholds = 250).

			WGD	NGD	BGD
wortbasiert	DEV	female	83.96	42.08	10.30
		male	112.98	52.89	11.35
	EVL	female	105.18	51.83	11.10
		male	95.49	46.82	10.67
phrasenbasiert	DEV	female	80.47	40.55	10.00
		male	106.33	50.21	10.99
	EVL	female	99.61	49.54	10.68
		male	90.01	44.57	10.33

3. Verbmobil–Korpus

Verbmobil ist ein vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie gefördertes Projekt zur maschinellen Übersetzung. Verbmobil (Phase I) erkennt gesprochene Sprache, analysiert die Eingabe, übersetzt sie ins Englische, erzeugt einen Satz und spricht ihn aus. Das System verarbeitet sprecherunabhängig Spontansprache (Szenario Terminabsprachen). Das in dieser Arbeit benutzte Korpus ist das Trainings– und Evaluierungskorpus des Erkenners für deutsche Spontansprache für die 1996er Auswertung (zuzüglich CD14) ohne Buchstabiereinheiten. Die Größe der Texte finden sich in Tabelle A.5.

Das akustische Vokabular besteht aus 5 328 Wörtern, zzgl. 35 Aussprachevarianten, 4 Häsitationen und 14 Geräuschmodellen, wobei die letzten beiden Ergänzungen in der Auswertung nicht berücksichtigt werden. Hinzu kommen acht Markierungen für unbekannte Wörter, das Satzende und sechs bedeutungslose Laute. Das akustische Modell wurde mit LDA und 6 317 (zzgl. Silencemodell) Triphonen modelliert, die mittels CART–Verfahren zu 2 000 + 1 tied states zusammengefasst und diese jeweils durch eine Gauß'sche Mischverteilungen dargestellt werden, wobei sich insgesamt 203 679 Einzelverteilungen ergeben. Die wort– und phrasenbasierten Wortgraphen für das aus 35 Dialogen mit 29 Sprechern bestehende Evaluierungskorpus wurden mit einer integrierten Trigramm–Suche erstellt. Dem Vokabular wurden im Rahmen

Tabelle A.4: Graph Error Rate (GER) für wortbasierte und phrasenbasierte Wortgraphen des NAB–Korpus (LMScale = 17, LM– und Lattice–Pruning–Thresholds = 250).

		Frauen GER[%]		Männer GER[%]		beide GER[%]	
		del/ins	tot	del/ins	tot	del/ins	tot
wortbasiert	DEV	0.1/0.6	3.7	0.2/0.7	4.4	0.2/0.6	4.1
	EVL	0.0/1.0	4.1	0.1/0.8	3.6	0.1/0.9	3.8
phrasenbasiert[a]	DEV	0.2/0.6	3.8	0.2/0.8	4.5	0.2/0.7	4.1
	EVL	0.0/1.0	4.1	0.1/0.7	3.7	0.1/0.8	3.9

[a]GER bezogen auf Wörter

Tabelle A.5: Größe des wort– bzw. phrasenbasierten Vokabulars, des Trainings– (Tr.) und des Evaluierungskorpus (Ev.) sowie der Out–of–Vocabulary–Rate (OOV), Verbmobil.

	Vokabular	Tr. (OOV[%])	Ev. (OOV[%])
wortbasiert	5 335	322 588 (1.1)	6 258 (1.5)
phrasenbasiert	5 435	286 252 (1.3)	5 517 (1.7)

Tabelle A.6: Word Graph Density (WGD), Node Graph Density (NGD), Bounds Graph Density (BGD) sowie Graph Error Rate (GER) für die Wortgraphen des Verbmobil–Evaluierungskorpus (LMScale = 15, LM– und Lattice–Pruning–Thresholds = 100).

	WGD	NGD	BGD	GER	
				del/ins	tot
wortbasiert	87.11	37.69	10.45	1.3/0.9	5.9
phrasenbasiert[a]	87.36	37.49	10.20	1.4/0.9	6.0

[a]GER bezogen auf Wörter

der Arbeit 200 Wortphrasen hinzugefügt, jedoch ohne das akustische Modell neu zu trainieren. Weitere Angaben zum Wortgraphen finden sich in Tabelle A.6.

Anhang B

Notation

In diesem Anhang wird die in den jeweiligen Kapiteln eingeführte Notation angegeben.

1. Sprachmodellierung

$Pr(\cdot)$	(unbekannte) tatsächliche Wahrscheinlichkeitsverteilung
$p_\theta(\cdot)$	modellierte Wahrscheinlichkeitsverteilung
θ	Parameter der modellierten Wahrscheinlichkeitsverteilung
x_1^T	Signalfolge der Länge T
w_1^N	Wortfolge der Länge N
N	Korpuslänge
n	Korpusposition
W	Vokabulargröße
t, u, v, w	Wörter
h	Worthistorie (Vorgängerwörter)
(h, w)	Verbundereignis: Wort w mit Worthistorie h
$N(\cdot)$	(absolute) Häufigkeit im Korpus
$\mathcal{L}_{w_1^N}(\theta)$	Likelihood der Wortfolge w_1^N für Parameter θ
$F_{w_1^N}(\theta)$	log–Likelihood der Wortfolge w_1^N für Parameter θ
PP	Perplexität

2. Glättungsverfahren

$\bar{h}$	verallgemeinerte Worthistorie
$\beta(w\|\bar{h})$	Glättungswahrscheinlichkeitsverteilung
λ	Interpolationsparameter (Glättungsparameter)
k	höchste beim Katz–Discounting noch geglättete Häufigkeit
d	(absoluter) Discountingparameter (Glättungsparameter)
r	absolute Häufigkeit eines Ereignisses
n_r	Anzahl r–mal gesehener Ereignisse

$n_1(\bar{h}, w)$	Anzahl Singletons (h, w) mit $h \in \bar{h}$ (d.h. alle Worthistorien h, für die $\bar{h}$ Verallgemeinerung ist)
$n_0(h)$	Anzahl der Wörter, die nie der Worthistorie h gefolgt sind

3. Varigramme

$h''=(v', h')$	die um das Vorgängerwort v' erweiterte Historie h'
$\Delta F(h'')$	Gewinn in der log–Likelihood durch Verwendung von Historie h''
H	Anzahl der Historien, um die das Ausgangs–Trigramm–Modell erweitert wurde

4. Wortphrasen

i	Anzahl Teilfolgen, in die eine Wortfolge w_1^N aufgeteilt wurde
j	j–te Teilfolge davon
l_j	Länge der j–ten Teilfolge
s_j	Position des ersten Wortes der j–ten Teilfolge in der Wortfolge w_1^N
π_j	Wörter der j–ten Teilfolge
$[w_m^n]$	Wortfolge von der Position m zur Position n, jeweils einschließlich, in der Wortfolge $w_1^N, m \leq n \leq N$
$Q(n)$	Wahrscheinlichkeit der Wortfolge $w_1^n,\ n \leq N$
$Q(n, l, l')$	Wahrscheinlichkeit der Wortfolge w_1^n, deren letzte beiden Teilfolgen die Längen l bzw. l' haben
$\tilde{i}(n)$	minimale Anzahl Phrasen bzw. Teilfolgen, die die Wortfolge w_1^N bis einschließlich Position n abdecken
a, b	Teilwörter einer Phrase
P	Anzahl Phrasen, um die das Vokabular erweitert worden ist

5. Wortklassen

$\mathcal{S}(w)$	Menge der Nachfolgerwörter zum Wort w im Trainingskorpus
$\mathcal{P}(w)$	Menge der Vorgängerwörter zum Wort w im Trainingskorpus
$\mathcal{S}(v, w)$	Menge der Nachfolgerwörter zum Wortpaar (v, w) im Trainingskorpus
$\mathcal{P}(v, w)$	Menge der Vorgängerwörter zum Wortpaar (v, w) im Trainingskorpus
G	Anzahl Wortklassen
$\mathcal{G}: w \rightarrow g_w$	Abbildung eines Wortes w auf seine Wortklasse g_w
$g,\ k$	Wortklassen
B	Anzahl unterschiedlicher Bigramme im Trainingskorpus
T	Anzahl unterschiedlicher Trigramme im Trainingskorpus

$F_{bi}(\mathcal{G})$	Bigramm–log–Likelihood
$F_{tri}(\mathcal{G})$	Trigramm–log–likelihood
I	Anzahl Iterationen des Clusteralgorithmus
$G(\cdot, w)$	$\sum_{g:N(g,w)>0} 1$ (Anzahl Vorgängerwortklassen des Wortes w)
$G(w, \cdot)$	$\sum_{g:N(w,g)>0} 1$ (Anzahl Nachfolgerwortklassen des Wortes w)
$\bar{G}_{\cdot w}$	$\frac{1}{W} \cdot \sum_{w} G(\cdot, w)$ (durchschnittliche Anzahl Vorgängerwortklassen)
$\bar{G}_{w\cdot}$	$\frac{1}{W} \cdot \sum_{w} G(w, \cdot)$ (durchschnittliche Anzahl Nachfolgerwortklassen)
$G(\cdot, \cdot, w)$	$\sum_{g_1,g_2:N(g_1,g_2,w)>0} 1$ (Anzahl Vorgängerwortklassenpaare des Wortes w)
$G(\cdot, w, \cdot)$	$\sum_{g_1,g_2:N(g_1,w,g_2)>0} 1$ (Anzahl Wortklassenpaare, die das Wort w einschließen)
$G(w, \cdot, \cdot)$	$\sum_{g_1,g_2:N(w,g_1,g_2)>0} 1$ (Anzahl Nachfolgerwortklassenpaare des Wortes w)
b	Discountparameter für Absolute Discounting
$N_r(g)$	Anzahl unterschiedlicher Wörter in Wortklasse g mit Häufigkeit r
$G_r(g_v, \cdot)$	Anzahl Nachfolgerwortklassen der Wortklasse g_v mit Häufigkeit r
$G_r(\cdot, g_w)$	Anzahl Vorgängerwortklassen der Wortklasse g_v mit Häufigkeit r
$G_r(\cdot, \cdot)$	Anzahl Wortklassenpaare mit Häufigkeit r

6. Abstand–n–gramme

$(t, \cdot, v, w)$, $(t, u, \cdot, w)$	Abstand–2–Trigramme
μ_1, μ_2, μ_3	Interpolationsparameter
$\alpha(t,u,v,w)$	allgemeines Gewicht für das Viergramm (t, u, v, w)
i	Feature (z. B. Trigramm (u, v, w) oder Abstand–Trigramm)
$f_i(h, w)$	Feature–Funktion des Feature i
e^{λ_i}	Gewicht eines Features i in einer log–linearen Verteilung
Λ	Menge der Gewichte (Parameter) einer log–linearen Verteilung
$p_\Lambda(w\|h)$	bedingte log–lineare Verteilung
$Z_\Lambda(h)$	Normierung einer log–lineare Verteilung für die Historie h
N_i	Häufigkeit des Features i im Trainingskorpus
$Q_i(\Lambda)$	erwartete Häufigkeit des Features i bezüglich einer log–linearen Verteilung mit Parametern Λ

Anhang C

Summenkriterium für Wortphrasen

In diesem Anhang wird die Berechnung der Hilfsgröße $Q(n,l,l')$ des Summenkriteriums aus Kapitel 5.1 zur Bestimmung der Satzwahrscheinlichkeit mittels Phrasen–Trigrammen durch Dynamische Programmierung hergeleitet (Gl. (5.1)).

$$
\begin{aligned}
Q(n,l,l') &= \sum_{i=1}^{n} \sum_{\substack{(l_1,\ldots,l_i) \\ l_i=l,\ l_{i-1}=l' \\ \pi_j \text{ existiert, } 1\le j\le i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{i=1}^{n-l} \sum_{\substack{(l_1,\ldots,l_i) \\ l_i=l' \\ \pi_j \text{ existiert, } 1\le j\le i}} p([w_{n-l+1}^{n}]|\pi_{i-1},\pi_i) \cdot \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{i=1}^{n-l} \sum_{l''=1}^{n-l-l'} \sum_{\substack{(l_1,\ldots,l_i) \\ l_i=l',\ l_{i-1}=l'' \\ \pi_j \text{ existiert, } 1\le j\le i}} p([w_{n-l+1}^{n}]|\pi_{i-1},\pi_i) \cdot \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{l''=1}^{n-l-l'} p([w_{n-l+1}^{n}]|[w_{n-l-l'-l''+1}^{n-l-l'}],[w_{n-l-l'+1}^{n-l}]) \cdot \sum_{i=1}^{n-l} \sum_{\substack{(l_1,\ldots,l_i) \\ l_i=l',\ l_{i-1}=l'' \\ \pi_j \text{ existiert, } 1\le j\le i}} \prod_{j=1}^{i} p(\pi_j|\pi_{j-2},\pi_{j-1}) \\
&= \sum_{l''=1}^{n-l-l'} p([w_{n-l+1}^{n}]|[w_{n-l-l'-l''+1}^{n-l-l'}],[w_{n-l-l'+1}^{n-l}]) \cdot Q(n-l,l',l'') \quad .
\end{aligned}
$$

Anhang D

Left–to–Right–Inside–Gleichung für Wortphrasen

Mit dem Summenkriterium aus Kapitel 5.1 ist die Gesamtwahrscheinlichkeit $p(w_1^N)$ einer Wortkette w_1^N durch Phrasen–Trigramme berechnet worden. In der Spracherkennung wird aber die bedingte Wahrscheinlichkeit $p(w_n|h_n)$ benötigt. Motiviert von [Jelinek et al. 90] ist daher für diese Arbeit eine Left–to–Right–Inside–Gleichung für Wortphrasen entwickelt worden, die in diesem Anhang beschrieben ist.

Statt der Wahrscheinlichkeit $Q(N) = p(w_1^N)$ einer abgeschlossenen Wortfolge w_1^N wird für eine beliebige Position n innerhalb dieser Wortfolge nun die Wahrscheinlichkeit $S(n) = p(w_1^n\_)$ des Präfixes $w_1^n\_$ benötigt. Da es sich um einen Präfix und nicht um eine abgeschlossene Wortfolge handelt, müssen nun auch diejenigen Wortphrasen berücksichtigt werden, die mit der Position n oder davor anfangen und über diese Position n hinausreichen. Es gilt dann

$$p(w_n|h_n) = \frac{p(w_1^n\_)}{\sum_{w'} p(w_1^{n-1}w'\_)} \quad . \tag{D.1}$$

Wegen

$$\sum_{w'} p(w_1^{n-1}w'\_) = \sum_{w'} p(w_1^{n-1}\_) = S(n-1)$$

läßt sich Gl. (D.1) effizient berechnen, da $S(n-1)$ aus der vorangegangenen Auswertung bekannt ist:

$$p(w_n|h_n) = \frac{S(n)}{S(n-1)} \quad .$$

$S(n)$ läßt sich aus $Q(n)$ bestimmen, indem ähnlich Gl. (5.1) für ein Trigramm–Modell die letzten drei Teilfolgen der Längen $l,\ l',\ l''$ betrachtet werden, wobei die letzte Teilfolge jedoch um m Wörter über die Position n hinausragt und sämtliche m–Tupel, die zusammen mit den $l-m$ gegebenen Wörtern bis einschließlich Position n eine Wortphrase bilden, berücksichtigt werden müssen:

$$S(n) =$$

$$= p(w_1^n \_)$$

$$= \sum_{l'',l'} \sum_{l} \sum_{m=0}^{l-1} \sum_{\substack{w'_1 \ldots w'_m \\ [w^n_{n-l+m+1} w'^m_1] \text{ existiert}}} p([w^n_{n-l+m+1} w'^m_1] | [w^{n-l-l'+m}_{n-l-l'-l''+m+1}][w^{n-l+m}_{n-l-l'+m+1}]) \cdot Q(n-l+m, l', l'') \quad .$$

Anhang E

Bigrammkriterium für die Wortphrasenauswahl

In Kapitel 5.2 ist ein auf Unigrammen basiertes Maximum–Likelihood–Kriterium zur Auswahl von Wortphrasen beschrieben worden. In diesem Anhang wird analog ein auf Bigrammen beruhendes Maximum–Likelihood–Kriterium beschrieben.

Hier wird die Verbesserung der Bigramm–log–Likelihood

$$\begin{aligned} F &= \sum_{v,\ w} N(v,w) \cdot \log p(w|v) \\ &= \sum_{v,\ w} N(v,w) \cdot \log \left[\frac{N(v,w)}{N(v)} \right] \end{aligned}$$

angestrebt. Werden die Wörter a und b zu einer Wortphrase c verschmolzen, so ändern sich bzw. entstehen die folgenden Häufigkeiten:

$$\begin{aligned} \tilde{N} &= N - N(a,b) \\ \\ \tilde{N}(a) &= N(a) - N(a,b) \\ \tilde{N}(b) &= N(b) - N(a,b) \\ \tilde{N}(c) &= N(a,b) \\ \\ \tilde{N}(a,b) &= 0 \\ \tilde{N}(v,a) &= N(v,a) - N(v,a,b) && v \neq b,c \\ \tilde{N}(b,w) &= N(b,w) - N(a,b,w) && w \neq a,c \\ \tilde{N}(b,a) &= N(b,a) - N(a,b,a) - N(b,a,b) + N(a,b,a,b) \end{aligned}$$

$$
\begin{array}{rcll}
\tilde{N}(v,c) & = & N(v,a,b) & v \neq c,b \\
\tilde{N}(b,c) & = & N(b,a,b) - N(a,b,a,b) & \\
\tilde{N}(c,w) & = & N(a,b,w) & w \neq c,a \\
\tilde{N}(c,a) & = & N(a,b,a) - N(a,b,a,b) & \\
\tilde{N}(c,c) & = & N(a,b,a,b) \quad . &
\end{array}
$$

Daraus ergibt sich die folgende Differenz in der log–Likelihood:

$\Delta F(a,b) =$

$$
\begin{aligned}
&= \sum_{v \neq c} \tilde{N}(v,c) \log \frac{\tilde{p}(c|v)}{p(a|v)p(b|a)} + \tilde{N}(c,c) \log \frac{\tilde{p}(c|c)}{p(a|b)p(b|a)} + \sum_{w \neq c} \tilde{N}(c,w) \log \frac{\tilde{p}(w|c)}{p(w|b)} \\
&\quad + \sum_{v \neq c,b} \tilde{N}(v,a) \log \frac{\tilde{p}(a|v)}{p(a|v)} + \tilde{N}(b,a) \log \frac{\tilde{p}(a|b)}{p(a|b)} + \sum_{w \neq c,a} \tilde{N}(b,w) \log \frac{\tilde{p}(w|b)}{p(w|b)} \\
&\quad + \sum_{w \neq c,b,a} \tilde{N}(a,w) \log \frac{\tilde{p}(w|a)}{p(w|a)}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{v \neq a,b} N(v,a,b) \log \frac{N(v,a,b)N(v)N(a)}{N(v)N(v,a)N(a,b)} \\
&\quad + [N(b,a,b) - N(a,b,a,b)] \log \frac{[N(b,a,b) - N(a,b,a,b)]\, N(b)N(a)}{[N(b) - N(a,b)]\, N(b,a)N(a,b)} \\
&\quad + N(a,a,b) \log \frac{N(a,a,b)N(a)N(a)}{[N(a) - N(a,b)]\, N(a,a)N(a,b)} \\
&\quad + N(a,b,a,b) \log \frac{N(a,b,a,b)N(b)N(a)}{N(a,b)N(b,a)N(a,b)} \\
&\quad + \sum_{w \neq a} N(a,b,w) \log \frac{N(a,b,w)N(b)}{N(a,b)N(b,w)} \\
&\quad + [N(a,b,a) - N(a,b,a,b)] \log \frac{[N(a,b,a) - N(a,b,a,b)]\, N(b)}{N(a,b)N(b,a)} \\
&\quad + \sum_{v \neq a,b} [N(v,a) - N(v,a,b)] \log \frac{[N(v,a) - N(v,a,b)]\, N(v)}{N(v)N(v,a)} \\
&\quad + [N(a,a) - N(a,a,b)] \log \frac{[N(a,a) - N(a,a,b)]\, N(a)}{[N(a) - N(a,b)]\, N(a,a)}
\end{aligned}
$$

$$+\sum_{w\neq a}\left[N(b,w)-N(a,b,w)\right]\log\frac{\left[N(b,w)-N(a,b,w)\right]N(b)}{\left[N(b)-N(a,b)\right]N(b,w)}$$

$$+\left[N(b,a)-N(a,b,a)-N(b,a,b)+N(a,b,a,b)\right]$$
$$\cdot\log\frac{\left[N(b,a)-N(a,b,a)-N(b,a,b)+N(a,b,a,b)\right]N(b)}{\left[N(b)-N(a,b)\right]N(b,a)}$$

$$+\sum_{w\neq a,b}N(a,w)\log\frac{N(a,w)N(a)}{\left[N(a)-N(a,b)\right]N(a,w)}$$

$$=N(a)\log N(a)+N(b)\log N(b)-2N(a,b)\log N(a,b)-N(b,a)\log N(b,a)$$
$$-\sum_{v\neq b}N(v,a)\log N(v,a)-\sum_{w\neq a}N(b,w)\log N(b,w)$$

$$-\left[N(a)-N(a,b)\right]\log\left[N(a)-N(a,b)\right]-\left[N(b)-N(a,b)\right]\log\left[N(b)-N(a,b)\right]$$
$$+\sum_{v\neq b}N(v,a,b)\log N(v,a,b)+\left[N(b,a,b)-N(a,b,a,b)\right]\log\left[N(b,a,b)-N(a,b,a,b)\right]$$
$$+N(a,b,a,b)\log N(a,b,a,b)$$
$$+\sum_{w\neq a}N(a,b,w)\log N(a,b,w)+\left[N(a,b,a)-N(a,b,a,b)\right]\log\left[N(a,b,a)-N(a,b,a,b)\right]$$
$$+\sum_{v\neq b}\left[N(v,a)-N(v,a,b)\right]\log\left[N(v,a)-N(v,a,b)\right]$$
$$+\sum_{w\neq a}\left[N(b,w)-N(a,b,w)\right]\log\left[N(b,w)-N(a,b,w)\right]$$
$$+\left[N(b,a)-N(a,b,a)-N(b,a,b)+N(a,b,a,b)\right]$$
$$\cdot\log\left[N(b,a)-N(a,b,a)-N(b,a,b)+N(a,b,a,b)\right]\quad.$$

Dieser komplexe Ausdruck läßt sich auch veranschaulicht darstellen als

$$\Delta F(a,b)=$$

$$=-\left[\sum_{v\neq b}N(v,a)\log p^*(v|a)+\sum_{w\neq a}N(b,w)\log p(w|b)+m(b,a)\right]$$
$$+\sum_{v\neq c,b}\tilde{N}(v,a)\log\tilde{p}^*(v|a)+\sum_{w\neq c,a}\tilde{N}(b,w)\log\tilde{p}(w|b)+\tilde{m}(b,a)$$
$$+\sum_{v\neq c,b}\tilde{N}(v,c)\log\tilde{p}^*(v|c)+\sum_{w\neq c,a}\tilde{N}(c,w)\log\tilde{p}(w|c)+\tilde{m}(c,a)+\tilde{m}(b,c)+\tilde{m}(c,c)$$

$$+N(b,a)\log\frac{N}{\tilde{N}}$$

mit

$$\begin{aligned}
p(a|b) &:= \frac{N(b,a)}{N(b)} && \text{Bigramm–Wahrscheinlichkeit}\\
p^*(b|a) &:= \frac{N(b,a)}{N(b)} && \text{rückwärtige Bigramm–Wahrscheinlichkeit}\\
m(b,a) &:= N(b,a)\log\frac{N(b,a)N}{N(b)N(a)} && \text{Mutual Information} \quad .
\end{aligned}$$

Anhang F

Leaving–One–Out–Kriterium für die Wortphrasenauswahl

Das in Kapitel 5.2 beschriebene Unigramm–log–Likelihood–Kriterium zur Phrasenauswahl soll in diesem Anhang durch Verwendung von Leaving–One–Out robuster gemacht werden, d.h. selten vorkommende Wortphrasen sollen durch dieses Kriterium möglichst nicht berücksichtigt werden.

Im Gegensatz zum einfachen Unigramm–log–Likelihood–Kriterium in Kap. 5.2 wird, ähnlich wie bei der Leaving–One–Out–Varigrammauswahl in Kap. 4.2, die aktuelle Beobachtung aus dem Trainingskorpus herausgenommen, d.h. die entsprechende Häufigkeit reduziert sich um eins. Singletons werden dadurch zu Zerotons, und eine Glättung muß stattfinden. In dieser Arbeit wird dazu Absolute Discounting mit Backing–Off und historienunabhängigem Discounting–Wert d verwendet:

$$\begin{aligned} F &= \sum_{w} N(w) \cdot \log p(w) \\ &= \sum_{w:\ N(w)>1} N(w) \cdot \log \left[\frac{N(w)-d-1}{N-1} \right] + n_1 \cdot \log \left[(n_+ - 1) \cdot \frac{d}{N-1} \cdot \frac{1}{n_0+1} \right] \end{aligned}$$

mit

$$\begin{aligned} n_1 &:= \sum_{w:\ N(w)=1} 1 \\ n_+ &:= \sum_{w:\ N(w)>0} 1 \\ n_0 &:= \sum_{w:\ N(w)=0} 1 \quad . \end{aligned}$$

Aus Gründen der Vereinfachung soll im Folgenden die realistische Annahme $\tilde{N}(a),\ \tilde{N}(b),\ \tilde{N}(c) > 1$ gelten. Durch Zusammenlegung der Wörter a und b zur Wortphrase c werden dieselben Zähler geändert wie im einfachen Unigramm–log–Likelihood–Kriterium, zusätzlich:

$$\begin{aligned}\tilde{n}_1 &= n_1 \\ \tilde{n}_+ &= n_+ + 1 \\ \tilde{n}_0 &= n_0 \quad .\end{aligned}$$

Dann gilt für die Differenz in der log–Likelihood:

$$\begin{aligned}
&\Delta F(a,b) = \\
&= \tilde{F}(a,b) - F \\
&= \tilde{N}(a) \cdot \log\left[\tilde{N}(a) - d - 1\right] + \tilde{N}(b) \cdot \log\left[\tilde{N}(b) - d - 1\right] + \tilde{N}(c) \cdot \log\left[\tilde{N}(c) - d - 1\right] \\
&\quad + \sum_{w \neq a,b,c:\ \tilde{N}(w)>1} \tilde{N}(w) \cdot \log\left[\tilde{N}(w) - d - 1\right] + \tilde{n}_1 \cdot \log\left[(\tilde{n}_+ - 1) \cdot d \cdot (\tilde{n}_0 + 1)^{-1}\right] \\
&\quad - \tilde{N} \cdot \log\left[\tilde{N} - 1\right] \\
&\quad - N(a) \cdot \log\left[N(a) - d - 1\right] - N(b) \cdot \log\left[N(b) - d - 1\right] \\
&\quad - \sum_{w \neq a,b:\ N(w)>1} N(w) \cdot \log\left[N(w) - d - 1\right] - n_1 \cdot \log\left[(n_+ - 1) \cdot d \cdot (n_0 + 1)^{-1}\right] \\
&\quad + N \cdot \log\left[N - 1\right] \\
&= \left[N(a) - N(a,b)\right] \cdot \log\left[N(a) - N(a,b) - d - 1\right] \\
&\quad + \left[N(b) - N(a,b)\right] \cdot \log\left[N(b) - N(a,b) - d - 1\right] \\
&\quad + N(a,b) \cdot \log\left[N(a,b) - d - 1\right] \\
&\quad - \left[N - N(a,b)\right] \cdot \log\left[N - N(a,b) - 1\right] \\
&\quad - N(a) \cdot \log\left[N(a) - d - 1\right] - N(b) \cdot \log\left[N(b) - d - 1\right] \\
&\quad + N \cdot \log\left[N - 1\right] \\
&\quad + n_1 \cdot \log\left[\frac{n_+}{n_+ - 1}\right] \quad .
\end{aligned}$$

Anhang G

Trigramm–log–Likelihood–Kriterium für Cluster–Algorithmus

In den Kapiteln 6.1 und 6.2 sind das log–Likelihood–Kriterium und ein entsprechender Cluster–Algorithmus auf Basis von Bigrammen vorgestellt worden. Im Rahmen dieser Arbeit wird in diesem Anhang dasselbe Kriterium auf Basis von Trigrammen hergeleitet. Die experimentellen Ergebnisse für dieses Trigramm–Kriterium finden sich in Kapitel 6.3.

Die Trigramm–log–Likelihood lautet

$$F(\mathcal{G}) = \sum_{w} N(w) \cdot \log\ p_0(w|g_w) + \sum_{g_u,g_v,g_w} N(g_u,g_v,g_w) \cdot \log\ p_2(g_w|g_u,g_v). \qquad \text{(G.1)}$$

mit der Maximum–Likelihood–Schätzung für die Transitionswahrscheinlichkeit

$$p_2(g_w|g_u,g_v) = \frac{N(g_u,g_v,g_w)}{N(g_u,g_v)} \quad . \qquad \text{(G.2)}$$

Mit Gl. (6.4) und Gl. (G.2) ergibt sich für die log–Likelihood–Gleichung (G.1):

$$\begin{aligned} F(\mathcal{G}) &= \sum_{u,v,w} N(u,v,w) \cdot \log p(w|u,v) \\ &= \sum_{w} N(w) \cdot \log \frac{N(w)}{N(g_w)} + \sum_{g_u,g_v,g_w} N(g_u,g_v,g_w) \cdot \log \frac{N(g_u,g_v,g_w)}{N(g_u,g_v)} \\ &= \sum_{g_u,g_v,g_w} N(g_u,g_v,g_w) \cdot \log N(g_u,g_v,g_w) - \sum_{g_u,g_v} N(g_u,g_v) \cdot \log N(g_u,g_v) \\ &\quad - \sum_{g_w} N(g_w) \cdot \log N(g_w) + \sum_{w} N(w) \cdot \log N(w) \quad . \end{aligned} \qquad \text{(G.3)}$$

Der Cluster–Algorithmus mit der Trigramm–log–Likelihood als Auswahlkriterium arbeitet prinzipiell genauso wie mit dem Bigramm–log–Likelihood–Kriterium in Kap. 6.2. Allerdings ändert sich durch Verwendung der Trigramm–log–Likelihood Gl. (G.3) auch

die effiziente Berechnung der Änderung der log–Likelihood durch Ausgliederung eines Wortes w aus seiner Wortklasse g_w bzw. durch Eingliederung des Wortes w in eine andere Wortklasse k. Wie in Kap. 6.2 sei auch hier die Ausgliederung betrachtet. Die Eingliederung verläuft analog.

Analog zur Bigramm–log–Likelihood sind durch eine Ausgliederung des Wortes w aus seiner Wortklasse g_w nur diejenigen Terme von Gl. (G.3) betroffen, in denen die Wortklasse g_w vorkommt.

- Einmaliges Vorkommen:

$$\forall g_1, g_2 \neq g_w : N(g_1, g_2, g_w) := N(g_1, g_2, g_w) - N(g_1, g_2, w) \quad , \tag{G.4}$$

  analog für $N(g_w, g_1, g_2)$ und $N(g_1, g_w, g_2)$.

- Zweimaliges Vorkommen:

$$\forall g \neq g_w : N(g, g_w, g_w) := N(g, g_w, g_w) - N(g, g_w, w) - N(g, w, g_w) + N(g, w, w) \quad ,$$

  analog für $N(g_w, g, g_w)$ und $N(g_w, g_w, g)$. Dies ist die Siebformel für zwei Mengen.

- Dreimaliges Vorkommen:

$$\begin{aligned} N(g_w, g_w, g_w) := N(g_w, g_w, g_w) - N(w, g_w, g_w) - N(g_w, g_w, w) - N(g_w, w, g_w) \\ + N(g_w, w, w) + N(w, g_w, w) + N(w, w, g_w) - N(w, w, w) \quad . \end{aligned}$$

  Dies ist die Siebformel für drei Mengen.

Die für die Trigramm–log–Likelihood notwendigen Hilfszähler ändern sich wie folgt:

- Keinmaliges Vorkommen von g_w: keine Änderung. Dies betrifft die Hilfszähler $N(g_1, g_2, w)$, $N(g_1, w, g_2)$, $N(w, g_1, g_2)$, $N(g, w, w)$, $N(w, g, w)$, $N(w, w, g)$ für $g_1,\ g_2,\ g \neq g_w$.

- Einmaliges Vorkommen von g_w:

$$\forall g \neq g_w : N(g, g_w, w) := N(g, g_w, w) - N(g, w, w) \quad ,$$

  analog für $N(g_w, g, w)$, $N(g, w, g_w)$, $N(g_w, w, g)$, $N(w, g, g_w)$, und $N(w, g_w, g)$.

$$N(g_w, w, w) := N(g_w, w, w) - N(w, w, w) \quad ,$$

  analog für $N(w, g_w, w)$ und $N(w, w, g_w)$.

- Zweimaliges Vorkommen von g_w:

$$N(g_w, g_w, w) := N(g_w, g_w, w) - N(g_w, w, w) - N(w, g_w, w) + N(w, w, w) \quad ,$$

  analog für $N(w, g_w, g_w)$ und $N(g_w, w, g_w)$. Dies ist wiederum die Siebformel für zwei Mengen.

Die Bigramm– und Unigrammhäufigkeiten ändern sich wie in Kap. 6.2 beschrieben.

Die Berechnung der Änderung der Trigramm–log–Likelihood ist nicht nur komplizierter, sondern auch sehr viel rechenaufwendiger, da in Gl. (G.4) und den beiden analogen Termen über zwei von g_w unabhängige Wortklassen aufsummiert wird anstatt über eine wie im Fall der Bigramm–log–Likelihood, Gl. (6.9) und Gl. (6.10). Damit steigt der Aufwand von grob $2 \cdot G$ zu $3 \cdot G^2$. Natürlich sind auch hier die meisten $N(g_1, g_2, w) = 0$, und die verfeinerte Berechnung mittels Listen der positiven Hilfsgrößen im Schritt 2 aus Kap. 6.2 ist hier noch wichtiger. Mit

$$G(\cdot, \cdot, w) := \sum_{g_1, g_2} N(g_1, g_2, w) > 0$$

und analog definierten Größen $G(\cdot, w, \cdot)$ und $G(w, \cdot, \cdot)$ ergibt sich ein Aufwand von $G(\cdot, \cdot, w) + G(\cdot, w, \cdot) + G(w, \cdot, \cdot)$ für Gl. (G.4) und die beiden analogen Terme.

Die Hilfszähler werden wie für die Bigramm–log–Likelihood in Kap. 6.2 für das aktuelle Wort w neu bestimmt. Da z.B. die Hilfsgröße $N(g_1, g_2, w) = \sum_{u \in g_1,\ v \in g_2} N(u, v, w)$ aus einer Summe über Worttrigramme besteht, sind für diese Worttrigramme zwei baumartige Trigrammstrukturen angelegt worden. In der einen Trigrammstruktur stehen die Trigramme in der Form (u, v, w) zur effizienten Bestimmung der Nachfolgerwörter $\mathcal{S}(u, v)$ zum Wortbigramm (u, v), in der anderen stehen die Trigramme in der Form (w, v, u) zur effizienten Bestimmung der Vorgängerwörter $\mathcal{P}(v, w)$ zum Wortbigramm (v, w). Insgesamt ergibt sich damit als Aufwandsabschätzung

$$I \cdot \sum_w \Bigg(\sum_{x \in \mathcal{S}(w)} |\mathcal{S}(w, x)| + \sum_{v \in \mathcal{P}(w)} |\mathcal{P}(v, w)| + \sum_{v \in \mathcal{P}(w)} |\mathcal{S}(v, w)| + G \cdot \Big(G(\cdot, \cdot, w) + G(\cdot, w, \cdot) + G(w, \cdot, \cdot) \Big) \Bigg) \quad .$$

Mit

$$\sum_w \sum_{x \in \mathcal{S}(w)} |\mathcal{S}(w, x)| = \sum_w \sum_{v \in \mathcal{P}(w)} |\mathcal{P}(v, w)| = \sum_w \sum_{v \in \mathcal{P}(w)} |\mathcal{S}(v, w)| = T$$

und $\bar{G}_{\cdot\cdot w}$, $\bar{G}_{\cdot w \cdot}$ und $\bar{G}_{w \cdot \cdot}$ als Mittelwerte für $G(\cdot, \cdot, w)$, $G(\cdot, w, \cdot)$ und $G(w, \cdot, \cdot)$ über alle Wörter w des Vokabulars ergibt sich:

$$\begin{aligned} & I \cdot \left(3 \cdot T + \sum_w G \cdot \Big(G(\cdot, \cdot, w) + G(\cdot, w, \cdot) + G(w, \cdot, \cdot) \Big) \right) \\ = \; & I \cdot \left(3 \cdot T + W \cdot G \cdot \left(\bar{G}_{\cdot\cdot w} + \bar{G}_{\cdot w \cdot} + \bar{G}_{w \cdot \cdot} \right) \right) \quad . \end{aligned} \tag{G.5}$$

Anhang H

Initiale Abbildungsfunktion für den Cluster–Algorithmus

Zur initialen Abbildung von Wörtern auf ihre Wortklassen aus Kapitel 6.2 sollen in diesem Anhang alternative Ansätze vorgestellt und experimentell verglichen werden.

Die Initialisierung kann bei lokal konvergenten Algorithmen wie den in Kap. 6.2 vorgestellten Cluster–Algorithmus einen großen Einfluß auf das Endergebnis haben. Es wurden deshalb drei unterschiedliche Ansätze untersucht:

baseline initialization: Dies ist die bereits in Kap. 6.2 verwendete Methode. Hier wird den häufigsten $G - 1$ Wörtern eine eigene Wortklasse zugewiesen, alle übrigen Wörter kommen in die verbleibende Wortklasse.

random initialization: Jedem Wort wird seine Wortklasse zufällig aufgrund einer Gleichverteilung zugewiesen.

POS initialization: Jedes Wort wird einer Wortklasse zugewiesen, die grob nach linguistischen Kriterien gebildet worden ist. Die Grundlage dazu war das Lexikon für das WSJ0–Korpus der Version 1.14 des regelbasierten Taggers von E. Brill [Brill 93], dessen Einträge mit linguistischen „Tags“ versehen sind, die ein Wort des Lexikons einer oder mehreren Parts–of–Speech–Wortmengen zuordnet. Da diese Zuordnung nicht eindeutig und darüberhinaus auch von der Groß– und Kleinschreibung abhängig ist, lassen sich diese Parts–of–Speech–Einteilungen nicht direkt auf die bisher verwendeten Wortklassen übertragen. Daher wurden die Tags eines Wortes mit unterschiedlicher Groß– und Kleinschreibung zusammengenommen. Weiter wurde in Tab. H.1 eine Hierarchie der Tags gebildet, bei der die wichtigen Tags, die auch nur kleine Wortmengen bilden, zuerst aufgelistet sind. Jedes Wort des Lexikons wird allein diesem seiner Tags zugeordnet, das in dieser Hierarchie an oberster Stelle steht. Da das Lexikon und das verwendete WSJ0–Vokabular nicht ganz übereinstimmen, wurden für die zusätzlichen Wörter des WSJ0–Vokabulars die Klassen 29–33 in Tab. H.1 eingerichtet und die Wörter diesen Wortklassen manuell zugeordnet. Die übrigen $G - 33$ Wortklassen blieben leer.

Tabelle H.1: Hierarchie der linguistischen Tags in Brills Lexikon.

1	**EX**	Existential *there*	18	**NNP**	Proper noun, singular
2	**TO**	*to*	19	**NNPS**	Proper noun, plural
3	**DT**	Determiner	20	**RB**	Adverb
4	**PRP**	Personal pronoun	21	**VB**	Verb, base form
5	**CD**	Cardinal number	22	**VBD**	Verb, past tense
6	**IN**	Preposition / subordinate conjunction	23	**VBG**	Verb, gerund / present participle
7	**PDT**	Predeterminer	24	**VBN**	Verb, past participle
8	**WP**	Wh–pronoun	25	**VBP**	Verb, non–3s, present
9	**WDT**	Wh–determiner	26	**VBZ**	Verb, 3s, present
10	**WRB**	Wh–adverb	27	**UH**	Interjection
11	**CC**	Coordinating conjunction	28	**FW**	Foreign word
12	**MD**	Modal	29		Auxiliary verb + *n't*
13	**JJ**	Adjective	30		PRP + ' + auxiliary verb
14	**JJR**	Comparative adjective	31		Possessive ending, singular
15	**JJS**	Superlative adjective	32		Possessive ending, plural
16	**NN**	Noun, singular or mass	33		Uncovered
17	**NNS**	Noun, plural			

Tabelle H.2: Perplexitäten auf Trainings– (PP_{Train}) und Testkorpora (PP_{Test}), WSJ0, und Anzahl der Iterationen I für $G = 500$ Wortklassen, Bigramm–log–Likelihood–Kriterium und verschiedenen initialen Abbildungsfunktionen $\mathcal{G} : w \rightarrow g_w$.

initiale Abbildungsfunktion		1M	4M	39M
baseline	I	20	22	32
	PP_{Train}	203.2	233.7	248.0
	PP_{Test}	326.8	259.9	244.2
random	I	23	35	21
	PP_{Train}	216.7	233.6	247.7
	PP_{Test}	319.9	260.7	243.4
POS	I	17	17	21
	PP_{Train}	203.5	233.8	248.3
	PP_{Test}	315.8	259.7	244.4

Die Versuche wurden für $G = 500$ Wortklassen und ausiteriertem Bigramm–log–Likelihood–Kriterium auf allen drei WSJ0–Korpora durchgeführt. Die resultierenden Testperplexitäten finden sich in Tab. H.2. Es zeigt sich, daß in den Ergebnissen kein nennenswerter Unterschied zwischen den initialen Abbildungsfunktionen besteht, lediglich die „POS initialization“ konvergiert etwas schneller als die anderen Ansätze. Entweder muß eine gute initiale Abbildungsfunktion noch gefunden werden, oder aber sie hat tatsächlich auf das Ergebnis des Cluster–Algorithmus keine Auswirkung.

Anhang I

Maximum Entropy mit hierarchischen Features

Für den Spezialfall hierarchischer Features (Trigramm, Bigramm) lassen sich die Constraint–Gleichungen (7.6) für die in Kapitel 7.1.2 vorgestellten Maximum–Entropy–Modelle geschlossen lösen. Diese Lösungen werden in diesem Anhang vorgestellt.

I.1 Ungeglättetes Trigramm–Modell

In diesem Anhang soll das Maximum–Entropy–Trigramm–Modell

$$p_\Lambda(w|u,v) = \frac{e^{\lambda_{uvw}+\lambda_{vw}+\lambda_w}}{Z_\Lambda(u,v)} \tag{I.1}$$

näher untersucht werden. Die Features dieses Modells werden als hierarchisch bezeichnet, da ein Auftreten des Trigramms (u,v,w) automatisch das Auftreten des Bigramms (v,w) und des Unigramms (w) zur Folge hat. Beispiel für nicht–hierarchische Features sind Abstand–Trigramme zusammen mit Trigrammen.

Für das Modell Gl. (I.1) läßt sich die Constraint–Gl. (7.6) für ein Trigramm–Feature (u,v,w) geschlossen lösen:

$$\frac{e^{\lambda_{uvw}+\lambda_{vw}+\lambda_w}}{Z_\Lambda(u,v)} = \frac{N(u,v,w)}{N(u,v)} \quad .$$

Damit ist das Maximum–Entropy–Trigramm–Modell nichts anderes als die ungeglättete log–Likelihood–Schätzung für Trigramme, also ihre relativen Häufigkeiten. Da sich die Wahrscheinlichkeiten der im Training beobachteten Trigramme bereits zu eins aufsummieren, bleibt trotz der Zufügung von Bi– und Unigrammen keine Wahrscheinlichkeitsmasse für die Glättung übrig. Damit gibt es auch für die Maximum–Entropy–Sprachmodellierung das Zero–Frequency–Problem. Vom rein mathematischen Standpunkt aus sind die Constraint–Gleichungen des Modells Gl. (I.1) sogar inkonsistent,

da es kein reelles λ mit $e^{\lambda} = 0$ gibt, wie sich aus $p_\Lambda(w|u,v) = 0$ im Falle ungesehener Trigramme ergibt.

I.2 Geglättetes Trigramm–Modell

Es ist kein naheliegendes Glättungsverfahren für Maximum–Entropy bekannt, deshalb wurden zwei bekannte Verfahren adaptiert [Ney 98]:

- **Cut–Offs:** Features i mit einer Häufigkeit $N_i \leq k$ werden weggelassen. Die Summe der N_i der weggelassenen Features wird für die ungesehenen Ereignisse reserviert.

- **Absolute Discounting:** Die Feature–Häufigkeiten N_i werden um einen festen Discountwert $0 < d < 1$ reduziert und die Summe der Reduktionen für die ungesehenen Ereignisse reserviert. Es gibt drei unterschiedliche Discountwerte, je einen für die Trigramm–, Bigramm– und Unigramm–Features. Mit der Reduktion der Feature–Häufigkeiten ändern sich auch die aus der log–Likelihood resultierenden Constraint–Gl. (7.6). Damit wird durch die geänderten Constraint–Gleichungen nicht mehr das Optimum der log–Likelihood beschrieben, und die Konvergenz des GIS ist nicht mehr garantiert.

Diese Verfahren lassen sich ohne weitere Betrachtung einfach implementieren. Hier ist aber von Interesse, welches Modell von den Constraint–Gleichungen mit diesen Glättungen beschrieben wird. Es sei dazu der Spezialfall betrachtet, daß alle Bigramme gesehen worden sind und deshalb nicht geglättet werden müssen, womit auch gleichzeitig die Unigramm–Features wegfallen. Dieser Fall ist zwar unrealistisch, führt aber zu einer geschlossenen Lösung. Bei der gleichzeitigen Anwendung beider beschriebener Glättungsverfahren, also sowohl Cut-Offs als auch Absolute Discounting, ergibt sich das Modell

$$p_\Lambda(w|u,v) = \begin{cases} \dfrac{e^{\lambda_{uvw}+\lambda_{vw}}}{Z_\Lambda(u,v)} & \text{if } N(u,v,w) > k \\ \dfrac{e^{\lambda_{vw}}}{Z_\Lambda(u,v)} & \text{otherwise} \end{cases} \tag{I.2}$$

und die Constraint–Gleichung für die Trigramme

$$\sum_{\tilde{u},\tilde{v},\tilde{w}} N(\tilde{u},\tilde{v}) \cdot p_\Lambda(\tilde{w}|\tilde{u},\tilde{v}) \cdot f_{uvw}(\tilde{u},\tilde{v},\tilde{w})$$
$$= N(u,v) \cdot \frac{e^{\lambda_{uvw}+\lambda_{vw}}}{Z_\Lambda(u,v)} = N(u,v,w) - d \quad ,$$

die sich auflösen läßt zu

$$\frac{e^{\lambda_{uvw}+\lambda_{vw}}}{Z_\Lambda(u,v)} = \frac{N(u,v,w)-d}{N(u,v)} \quad . \tag{I.3}$$

Gl. (I.3) wird für die Lösung der Constraint–Gleichung für Bigramme benötigt:

$$\sum_{\tilde{u},\tilde{v},\tilde{w}} N(\tilde{u},\tilde{v}) \cdot p_\Lambda(\tilde{w}|\tilde{u},\tilde{v}) \cdot f_{vw}(\tilde{u},\tilde{v},\tilde{w})$$
$$= \sum_{\tilde{u}:\ N(\tilde{u},v,w)\le k} N(\tilde{u},v) \cdot \frac{e^{\lambda_{vw}}}{Z_\Lambda(\tilde{u},v)} + \sum_{\tilde{u}:\ N(\tilde{u},v,w)>k} N(\tilde{u},v,w) - d = N(v,w) \quad ,$$

die sich auflösen läßt zu

$$e^{\lambda_{vw}} = \frac{N_{\le k}(\cdot,v,w) + n_{>k}(\cdot,v,w)\cdot d}{\sum_{\tilde{u}:\ N(\tilde{u},v,w)\le k} \frac{N(\tilde{u},v)}{Z_\Lambda(\tilde{u},v)}}$$
$$\approx \frac{N_{\le k}(\cdot,v,w) + n_{>k}(\cdot,v,w)\cdot d}{\sum_{\tilde{u}} \frac{N(\tilde{u},v)}{Z_\Lambda(\tilde{u},v)}} \quad , \tag{I.4}$$

mit

$$N_{\le k}(\cdot,v,w) := \sum_{\tilde{u}:\ N(\tilde{u},v,w)\le k} N(\tilde{u},v,w)$$
$$n_{>k}(\cdot,v,w) := \sum_{\tilde{u}:\ N(\tilde{u},v,w)>k} 1 \quad .$$

Die Näherung scheint gerechtfertigt, da auf den verwendeten Korpora fast alle möglichen Trigramme nicht gesehen worden sind. Eine exakte Abschätzung des Fehlers dieser Näherung ist aber in dieser Arbeit nicht gelungen. Die experimentellen Ergebnisse bestätigen jedoch, daß dieser Fehler klein ist. Die Berechnung der Renormierung $Z_\Lambda(u,v)$ führt, ebenfalls unter Verwendung von Gl. (I.3), zu

$$Z_\Lambda(u,v) = \frac{\sum_{\tilde{w}:\ N(u,v,\tilde{w})\le k} e^{\lambda_{v\tilde{w}}}}{\frac{N_{\le k}(u,v,\cdot) + n_{>k}(u,v,\cdot)\cdot d}{N(u,v)}} \quad , \tag{I.5}$$

mit

$$N_{\leq k}(u,v,\cdot) := \sum_{\tilde{w}:\ N(u,v,w)\leq k} N(u,v,\tilde{w})$$

$$n_{>k}(u,v,\cdot) := \sum_{\tilde{w}:\ N(u,v,\tilde{w})>k} 1 \quad .$$

Dabei ist

$$\frac{N_{\leq k}(u,v,\cdot) + n_{>k}(u,v,\cdot)\cdot d}{N(u,v)}$$

die Wahrscheinlichkeitsmasse zur Verteilung über die ungesehenen Ereignisse für eine Historie $h = (u,v)$. Die Einsetzung von Gl. (I.3), Gl. (I.4) und Gl. (I.5) in das Sprachmodell Gl. (I.2) liefert das resultierende Sprachmodell. Zur besseren Übersichtlichkeit und Vergleichbarkeit mit bekannten Glättungsverfahren ist hier wieder zwischen Cut–Offs und Absolute Discounting unterschieden.

- **Cut–Offs** ($d = 0$, $k > 0$):

$$p_\Lambda(w|u,v) = \begin{cases} \dfrac{N(u,v,w)}{N(u,v)} & \text{if } N(u,v,w) > k \\ \dfrac{N_{\leq k}(u,v,\cdot)}{N(u,v)} \cdot \beta_{uv}(w) & \text{otherwise} \end{cases}$$

  mit

$$\beta_{uv}(w) := \frac{N_{\leq k}(\cdot,v,w)}{\sum_{\tilde{w}:\ N(u,v,w)\leq k} N_{\leq k}(\cdot,v,\tilde{w})} \quad .$$

  Damit handelt es sich hier um eine bereits aus Kap. 3 bekannte Backing–Off–Glättung, allerdings mit einer bisher unbekannten Glättungsverteilung $\beta_{uv}(w)$, die für den Fall

$$\sum_{\tilde{w}:\ N(u,v,w)\leq k} N_{\leq k}(\cdot,v,\tilde{w}) = 0$$

  mangels Wahrscheinlichkeitsmasse nicht definiert ist. Es ist nicht klar, wie Generalized Iterative Scaling mit diesem Problem umgeht.

- **Absolute Discounting** ($0 < d < 1$, $k = 0$):

$$p_\Lambda(w|u,v) = \begin{cases} \dfrac{N(u,v,w) - d}{N(u,v)} & \text{if } N(u,v,w) > 0 \\ \dfrac{n_{>0}(u,v,\cdot)\cdot d}{N(u,v)} \cdot \beta_{uv}(w) & \text{otherwise} \end{cases}$$

  mit

$$\beta_{uv}(w) := \frac{n_{>0}(\cdot,v,w)}{\sum_{\tilde{w}:\ N(u,v,\tilde{w})=0} n_{>0}(\cdot,v,\tilde{w})} \quad .$$

  Damit handelt es sich hier ebenfalls um das bekannte Backing–Off–Verfahren mit einer Glättungsverteilung $\beta_{uv}(w)$, die als „Marginal Distribution“ [Kneser & Ney 95] bekannt ist und der in dieser Arbeit verwendeten Singleton–Glättungsverteilung ähnelt.

Tabelle I.1: Testperplexitäten für geglättete relative Trigramm–Häufigkeiten und Maximum–Entropy–Trigramm–Modell mit verschiedenen Glättungsverfahren auf dem WSJ0–4M Korpus.

Modell	PP
geglättete relative Häufigkeiten:	
backing–off	163.4
backing–off, marginal distribution	153.2
interpolation, singleton distribution	152.1
Maximum–Entropy (20 Iterationen):	
Cut–Offs, $k = 1$	178.7
Absolute Discounting	157.9

Dieser enge Zusammenhang zwischen Maximum–Entropy–Sprachmodellen und gängigen Verfahren zur Glättung relativer Häufigkeiten war bisher nicht bekannt.

Die analytische Beschreibung wird durch experimentelle Ergebnisse unterlegt. In Tabelle I.1 werden geglättete relative Häufigkeiten bei verschiedenen Glättungsmethoden mit dem Maximum–Entropy–Trigramm Gl. (I.1) auf WSJ0–4M verglichen. Tatsächlich ist die Perplexität für Maximum–Entropy mit Absolute Discounting deutlich niedriger als für geglätte relative Häufigkeiten mit Backing–Off, Absolute Discounting und relativen Häufigkeiten als Glättungsverteilung. Es erreicht jedoch nicht die Perplexität der explizit modellierten Marginal–Glättungsverteilung. Das übliche Trigramm, mit Interpolation anstelle von Backing–Off, ist nochmals etwas besser. Glättung durch Cut–Offs hingegen ist deutlich schlechter als alle übrigen Methoden. Durch das Weglassen der seltenen Features wird das Sprachmodell offenbar zu stark vergröbert.

Anhang J

Beschleunigte Berechnung des Erwartungswertes für Maximum Entropy

In Ergänzung zur schnellen Berechnung der Renormierung $Z_\Lambda(h)$ nach [Stolcke et al. 97] ist im Rahmen dieser Arbeit auch eine schnelle Berechnung des Erwartungswertes $Q_i(\Lambda)$ für das Training der Maximum–Entropy–Modelle aus Kapitel 7.1.2 mittels Generalized Iterative Scaling hergeleitet worden. Diese schnelle Berechnung wird in diesem Anhang vorgestellt.

In der vorliegenden Implementierung wird der Erwartungswert $Q_i(\Lambda)$ korpusbasiert berechnet, d.h. als

$$\begin{aligned} Q_i(\Lambda) &= \sum_{hw} N(h) \cdot p_\Lambda(w|h) \cdot f_i(h,w) \\ &= \sum_{n=1}^{N} \sum_{w} p_\Lambda(w|h_n) \cdot f_i(h_n,w) \quad . \end{aligned} \tag{J.1}$$

berechnet. Ist i ein Unigramm–Feature, dann geht nach Gl. (J.1) jede Korpusposition n in die Berechnung von $Q_i(\Lambda)$ ein. Damit sind für die Unigramm–Features $W \cdot N$ Berechnungen nötig. Um diesen Aufwand deutlich zu reduzieren, wird ähnlich wie in [Stolcke et al. 97] die Features in historienabhängige („conditional“) Features $i \in I_c$, z.B. Bigramm– und Trigramm–Features, und historienunabhängige („marginal“) Features $i \in I_m$, z.B. Unigramm–Features, unterteilt:

$$\begin{aligned} I_m &:= \{ i \in I \mid \forall h, h' : f_i(h,w) = f_i(h',w) \} \\ I_c &:= \{ i \in I \mid \exists h, h' : f_i(h,w) \neq f_i(h',w) \} \quad , \end{aligned}$$

wobei jedes Feature nur einer der beiden Mengen angehören kann. Darauf aufbauend wird zu jeder Historie h eine Liste $W(h)$ der Wörter w geführt, für die es ein historienabhängiges Feature $i \in (h,w)$ gibt:

$$W(h) := \{ w \in W \mid \exists i \in I_c : f_i(h,w) \neq 0 \} \quad .$$

Mit Hilfe der Wortliste $W(h)$ läßt sich der Aufwand der Berechnung wie folgt vereinfachen:

$$
\begin{aligned}
Q_w(\Lambda) &= \sum_{n=1}^{N} p_\Lambda(w\,|\,h_n) && \text{(J.2)}\\
&= \sum_{n:w\in W(h_n)} p_\Lambda(w\,|\,h_n) + \sum_{n:w\notin W(h_n)} p_\Lambda(w\,|\,h_n)\\
&= \sum_{n:w\in W(h_n)} p_\Lambda(w\,|\,h_n) + \sum_{n:w\notin W(h_n)} \frac{e^{\lambda_w}}{Z_\Lambda(h_n)}\\
&= \sum_{n:w\in W(h_n)} p_\Lambda(w\,|\,h_n) + \sum_{n=1}^{N} \frac{e^{\lambda_w}}{Z_\Lambda(h_n)} - \sum_{n:w\in W(h_n)} \frac{e^{\lambda_w}}{Z_\Lambda(h_n)}\\
&= e^{\lambda_w} \sum_{n=1}^{N} \frac{1}{Z_\Lambda(h_n)} + \sum_{n:w\in W(h_n)} \left[p_\Lambda(w\,|\,h_n) - \frac{e^{\lambda_w}}{Z_\Lambda(h_n)} \right]. && \text{(J.3)}
\end{aligned}
$$

Mit den Definitionen

$$
\begin{aligned}
Z'_\Lambda &:= \sum_{n=1}^{N} \frac{1}{Z_\Lambda(h_n)}\\
Q'_w(\Lambda) &:= \sum_{n:w\in W(h_n)} \left[p_\Lambda(w\,|\,h_n) - \frac{e^{\lambda_w}}{Z_\Lambda(h_n)} \right]
\end{aligned}
$$

läßt sich Gl. (J.3) schreiben als

$$
Q_w(\Lambda) = e^{\lambda_w} \cdot Z'_\Lambda + Q'_w(\Lambda) \quad .
$$

Die Berechnung der Größe Z'_Λ erfordert keinen Mehraufwand, da die Renormierung $Z_\Lambda(h_n)$ ohnehin an jeder Korpusposition n berechnet werden muß. Die Größe $Q'_w(\Lambda)$ muß im Gegensatz zur naiven Implementierung Gl. (J.2) nur die Korpuspositionen n berücksichtigen, an denen das Wort w durch ein historienabhängiges Feature betroffen sein kann. Durch die effiziente Berechnung verringert sich der Aufwand somit von $N \cdot W$ zu $\sum_n W(h_n) = N \cdot W_c$ mit W_c als der durchschnittlichen Anzahl Wörter pro Korpusposition, die durch ein historienabhängiges Feature betroffen sein kann. Da in den verwendeten Korpora nur ein sehr kleiner Teil der möglichen Bigramme und Trigramme im Training gesehen worden ist, gilt somit $W_c \ll W$.

Literaturverzeichnis

[Bahl et al. 83] Bahl, L. R., Jelinek, F., Mercer, R. L.: "A Maximum Likelihood Approach to Continuous Speech Recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 5, pp. 179–190, 1983.

[Baum 72] L. E. Baum: "An Inequality and Associated Maximization Technique in Statistical Estimation of a Markov Process", Inequalities, Vol. 3, No. 1, pp. 1–8, 1972.

[Berger et al. 94] A. Berger, P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, J. Gillett, J. Lafferty, R. Mercer, H. Printz, L. Ures: "The Candide System for Machine Translation", ARPA Human Language Technology Workshop, Plainsboro, NJ, pp. 152–157, März 1994.

[Berger et al. 96] A. L. Berger, S. Della Pietra, V. Della Pietra: "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics, Vol. 22, No. 1, pp. 39–71, 1996.

[Bimbot et al. 95] F. Bimbot, R. Pieraccini, E. Levin, B. Atal: "Variable–Length Sequence Modeling: Multigrams", IEEE Signal Processing Letters, Vol. 2, No. 6, Juni 1995.

[Binney et al. 92] J. Binney, N. Dowrick, A. Fisher, M. Newman: "The Theory of Critical Phenomena", Oxford University Press, Oxford, 1992.

[Bishop et al. 75] Y. M. M. Bishop, S. E. Fienberg, P. W. Holland: "Discrete Multivariate Analysis", MIT Press, Cambridge, MA, 1975.

[Brill 93] E. Brill: "A Corpus-Based Approach to Language Learning", PhD thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1993.

[Brown et al. 90] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, P. Roossin: "A Statistical Approach to Machine Translation", Computational Linguistics, Vol. 16.2, pp. 79-85, 1990.

[Brown et al. 92] P. Brown, V. Della Pietra, P. de Souza, J. Lai, R. Mercer: "Class–Based n–gram Models of Natural Language", Computational Linguistics, Vol. 18, No. 4, pp. 467–479, 1992.

[Chelba et al. 97] C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, D. Wu: "Structure and Performance of a Dependency Language Model", 5th European Conference on Speech Communication and Technology, Rhodos, pp. 2775–2778, 1997.

[Chelba & Jelinek 98] C. Chelba, F. Jelinek: "Exploiting Syntactic Structure for Language Modeling", 36th Annual Meeting of the Association for Computational Linguistics, Montreal, Quebec, pp. 225–231, August 1998.

[Collins 96] M. Collins: "A New Statistical Parser Based on Bigram Lexical Dependencies", 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, pp. 184–191, Juni 1996.

[Collins 97] M. Collins: "Three Generative, Lexicalised Models for Statistical Parsing", 35th Annual Meeting of the Association for Computational Linguistics, Madrid, pp. 16–23, Juli 1997.

[Darroch & Ratcliff 72] J. N. Darroch, D. Ratcliff: "Generalized Iterative Scaling for Log–Linear Models", Annals of Mathematical Statistics, Vol. 43, pp. 1470–1480, 1972.

[Deligne & Bimbot 95] S. Deligne, F. Bimbot: "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", IEEE International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, Vol. 1, pp. 169–172, Mai 1995.

[Della Pietra et al. 94] S. Della Pietra, V. Della Pietra, J. Gillet, J. Lafferty, H. Printz, L. Ureš: "Inference and Estimation of a Long–Range Trigram Model", in: R. C. Carrasco, J. Oncina (eds.): "Grammatical Inference and Applications", Second International Colloquium, ICGI–94, Alicante, Spanien, (Springer Lecture Notes in Artificial Intelligence No. 862, Springer–Verlag, Berlin), pp. 78–92, 1994.

[Della Pietra et al. 95] S. Della Pietra, V. Della Pietra, J. Lafferty: "Inducing Features of Random Fields", Technical Report CMU-CS-95-144, Carnegie Mellon University, Pittsburgh, PA, 1995.

[Dempster et al. 77] A. P. Dempster, N. M. Laird, D. B. Rubin: "Maximum Likelihood from Incomplete Data via the EM Algorithm", J. Royal Statist. Soc. Ser. B (methodological), Vol. 39, pp. 1–38, 1977.

[Duda & Hart 73] R. O. Duda, P. E. Hart: "Pattern Classification and Scene Analysis", J. Wiley & Sons, NY, 1973.

[Engesser 94] H. Engesser: "Duden Rechnen und Mathematik", 5. Auflage, Dudenverlag, Mannheim, 1994.

[Gauvain et al. 97] J. L. Gauvain, G. Adda, L. Lamel, M. Adda–Decker: "Transcribing Broadcast News: The LIMSI Nov96 Hub4 System", DARPA Speech Recognition Workshop, Chantilly, VA, pp. 56–63, Februar 1997.

[Generet et al. 95] M. Generet, H. Ney, F. Wessel: "Extensions of Absolute Discounting for Language Modeling", 4th European Conference on Speech Communication and Technology, Madrid, pp. 1245–1248, September 1995.

[Giachin 95] E. Giachin: "Phrase Bigrams for Continuous Speech Recognition", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, pp. 225–228, Mai 1995.

[Giachin et al. 94] E. Giachin, P. Baggia, G. Micca: "Language Models for Spontaneous Speech Recognition: A Bootstrap Method for Learning Phrase Bigrams", International Conference on Spoken Language Processing, Yokohama, pp. 843–846, September 1994.

[Good 53] I. J. Good: "The Population Frequencies of Species and the Estimation of Population Parameters", Biometrika, Vol. 40, pp. 237-264, Dezember 1953.

[Hwang 97] K. Hwang: "Vocabulary Optimization Based on Perplexity", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, München, Vol. II, pp. 1419–1422, April 1997.

[Iyer & Ostendorf 96] R. Iyer, M. Ostendorf: "Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models", Fourth International Conference on Spoken Language Processing, Philadelphia, PA, Vol. 1, pp. 236–239, Oktober 1996.

[Jardino 96] M. Jardino: "Multilingual Stochastic n-Gram Class Language Models", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, pp. 161-163, Mai 1996.

[Jardino & Adda 93] M. Jardino, G. Adda: "Automatic Word Classification Using Simulated Annealing", Proc. 3rd European Conference on Speech Communication and Technology, Berlin, pp. 1191–1194, 1993.

[Jardino & Adda 94] M. Jardino, G. Adda: "Automatic Determination of a Stochastic Bi–Gram Class Language Model", in: R. C. Carrasco, J. Oncina (eds.): "Grammatical Inference and Applications", Second International Colloquium, ICGI–94, Alicante, Spanien, (Springer Lecture Notes in Artificial Intelligence No. 862, Springer–Verlag, Berlin), pp. 57–65, 1994.

[Jaynes 57] E. T. Jaynes: "Information Theory and Statistical Mechanics", Physics Reviews, Vol. 106, pp. 620–630, 1957.

[Jelinek 91] F. Jelinek: "Self-Organized Language Modeling for Speech Recognition", in: A. Waibel and K.-F. Lee (eds.): "Readings in Speech Recognition", (Morgan Kaufmann Publishers, San Mateo, CA), pp. 450-506, 1991.

[Jelinek et al. 90] F. Jelinek, J. Lafferty, R. L. Mercer: "Basic methods of probabilistic context–free grammars", in: P. Laface and R. DeMori (eds.): "Speech Recognition and Understanding. Recent Advances, Trends and Applications", Proceedings of the NATO Advanced Study Institute, Cetraro, Italy, pp. 345–360, Juli 1990 (Vol. F–75 of NATO Advanced Study Institute Series, Springer–Verlag, Berlin, 1992).

[Jelinek & Mercer 80] F. Jelinek, R. L. Mercer: "Interpolated Estimation of Markov Source Parameters from Sparse Data", in: E. S. Gelsema, L. N. Kanal (eds.): "Pattern Recognition in Practice", North Holland, Amsterdam, pp. 381–397, 1980.

[Jelinek et al. 91] F. Jelinek, B. Merialdo, S. Roukos, M. Strauss: "A Dynamic Language Model for Speech Recognition", DARPA Speech and Natural Language Workshop, Pacific Grove, CA, pp. 293–295, Februar 1991.

[Katz 87] S. M. Katz: "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 35, pp. 400-401, März 1987.

[Klakow 98a] D. Klakow: "Language–Model Optimization by Mapping of Corpora", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, WA, Vol. II, pp. 701–704, Mai 1998.

[Klakow 98b] D. Klakow: "Log-linear Interpolation of Language Models", Int. Conf. on Speech and Language Processing, Sydney, Vol. 5, pp. 1695–1699, Dezember 1998.

[Klakow et al. 98] D. Klakow, X. Aubert, P. Beyerlein, R. Haeb-Umbach, M. Ullrich, A. Wendemuth, P. Wilcox: "Language–Model Investigations Related to Broadcast News", DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, pp. 265–269, Februar 1998.

[Kneser 96] R. Kneser: "Statistical Language Modeling Using a Variable Context Length", Fourth International Conference on Spoken Language Processing, Philadelphia, PA, Vol. 1, pp. 494–497, Oktober 1996.

[Kneser & Ney 93] R. Kneser, H. Ney: "Improved Clustering Techniques for Class–Based Statistical Language Modelling", 3rd European Conference on Speech Communication and Technology, Berlin, pp. 973–976, 1993.

[Kneser & Ney 95] R. Kneser, H. Ney: "Improved Backing-Off for m–gram Language Modeling", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, Vol. I, pp. 49-52, Mai 1995.

[Kneser & Peters 97] R. Kneser, J. Peters: "Semantic Clustering for Adaptive Language Modeling", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, München, pp. 779–782, April 1997.

[Landau & Lifschitz 87] L. Landau, E. Lifschitz: "Statistische Physik", Teil I, Akademie–Verlag, Berlin (Ost), 1987.

[Lehmann 83] E. L. Lehmann: "Theory of Point Estimation", J. Wiley, New York, 1983.

[Levenshtein 66] V. Levenshtein: "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals", Soviet Physics — Doklady, Vol. 10, pp. 707–710, Februar 1966.

[Masataki & Sagisaka 96] H. Masataki, Y. Sagisaka: "Variable–Order N–Gram Generation by Word–Class Splitting and Consecutive Word Grouping", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, pp. 188–191, Mai 1996.

[McCandless & Glass 93] M. K. McCandless, J. R. Glass: "Empirical Acquisition of Word and Phrase Classes in the ATIS Domain", 3rd European Conference on Speech Communication and Technology, Berlin, Vol. 2, pp. 981–984, September 1993.

[McCandless & Glass 94] M. K. McCandless, J. R. Glass: "Empirical Acquisition of Language Models for Speech Recognition", International Conference on Spoken Language Processing, Yokohama, pp. 835–838, September 1994.

[Metropolis et al. 53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller: "Equation of State Calculations by Fast Computing Machines", Journal of Chemical Physics, Vol. 21, No. 6, pp. 1087–1092, 1953.

[Ney 92] H. Ney: "Stochastic Grammars and Pattern Recognition", in: P. Laface and R. DeMori (eds.): "Speech Recognition and Understanding. Recent Advances, Trends and Applications", Proceedings of the NATO Advanced Study Institute, Cetraro, Italy, pp. 319–344, Juli 1990 (Vol. F–75 of NATO Advanced Study Institute Series, Springer–Verlag, Berlin, 1992).

[Ney 94] H. Ney: Persönlicher Meinungsaustausch, RWTH Aachen, 1994.

[Ney 98] H. Ney: Persönlicher Meinungsaustausch, RWTH Aachen, 1998.

[Ney et al. 94] H. Ney, U. Essen, R. Kneser: "On Structuring Probabilistic Dependences in Stochastic Language Modelling", Computer Speech and Language, Vol. 8, pp. 1–38, 1994.

[Ney et al. 95] H. Ney, U. Essen, R. Kneser: "On the Estimation of Small Probabilities by Leaving–One–Out", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-17, No. 12, pp. 1202–1212, 1995.

[Niesler 97] T. R. Niesler: "Category–Based Statistical Language Models", PhD Thesis, University of Cambridge, Juni 1997.

[Niesler & Woodland 96] T. R. Niesler, P. C. Woodland: "A Variable–length Category–based N–gram Language Model", IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, Vol. 1, pp. 164–167, Mai 1996.

[Ortmanns et al. 97] S. Ortmanns, H. Ney, X. Aubert: "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition", Computer, Speech, and Language, Vol. 11, No. 1, pp. 43–72, Januar 1997.

[Pereira et al. 96] F. Pereira, Y. Singer, N. Tishby: "Beyond Word N–Grams", unveröffentlichter Bericht, März 1996.

[Riccardi et al. 97] G. Riccardi, A. L. Gorin, A. Ljolje, M. Riley: "A Spoken Language System for Automated Call Routing", IEEE International Conference on Acoustics, Speech, and Signal Processing, München, Vol. 2, pp. 1143–1146, April 1997.

[Ries et al. 96] K. Ries, F. D. Buø, A. Waibel: "Class Phrase Models for Language Modeling", Fourth International Conference on Spoken Language Processing, Philadelphia, PA, pp. 398–401, Oktober 1996.

[Ron et al. 94] D. Ron, Y. Singer, N. Tishby: "The Power of Amnesia", in: J. Cowan et al. (eds.): "Advances in Neural Information Processing Systems", Vol. 6, Morgan Kauffmann, San Mateo, CA, pp. 176–183, 1994.

[Rosenfeld 94] R. Rosenfeld: "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", Ph.D. Thesis, Technical Report CMU-CS-94-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994.

[Sankar et al. 98] A. Sankar, F. Weng, Z. Rivlin, A. Stolcke, R. Gadde: "The Development of SRI's 1997 Broadcast News Transcription System", DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, pp. 91–96, Februar 1998.

[Seymore et al. 97] K. Seymore, S. Chen, M. Eskenazi, R. Rosenfeld: "Language and Pronunciation Modeling in the CMU 1996 Hub 4 Evaluation", DARPA Speech Recognition Workshop, Chantilly, VA, pp. 141–146, Februar 1997.

[Seymore et al. 98] K. Seymore, S. Chen, S. Doh, M. Eskenazi, E. Gouvêa, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern, E. Thayer: "The 1997 CMU Sphinx–3 English Broadcast News Transcription System", DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, pp. 55–59, Februar 1998.

[Stolcke et al. 97] A. Stolcke, C. Chelba, D. Engle, V. Jimenez, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, D. Wu, F. Jelinek, S. Khudanpur: "Dependency Language Modeling", 1996 Large Vocabulary Continuous Speech Recognition Summer Research Workshop Technical Reports, Research Note 24, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, April 1997.

[Tillmann & Ney 96] C. Tillmann, H. Ney: "Selection Criteria for Word Trigger Pairs in Language Modeling", Third Int. Colloquium on Grammatical Inference, Montpellier, pp. 95–106, September 1996.

[Tillmann & Ney 97] C. Tillmann, H. Ney: "Word Trigger and the EM Algorithm", ACL Computational Natural Language Learning Workshop, Madrid, pp. 117–124, Juli 1997.

[Tillmann et al. 97] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, H. Sawaf: "Accelerated DP Based Search for Statistical Translation", 5th European Conference on Speech Communication and Technology, Rhodos, pp. 2667–2670, 1997.

[Woodland et al. 98] P. Woodland, T. Hain, S. Johnson, T. Niesler, A. Tuerk, E. Whittaker, S. Young: "The 1997 HTK Broadcast News Transcription System", DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, pp. 41–48, Februar 1998.

Lebenslauf

Sven Carl Martin, geb. 28. April 1966 in Köln

Schulbesuch

1972 bis 1976	Gemeinschaftsgrundschule Kerpen–Süd
1976 bis 1985	Tagesheimgymnasium der Stadt Kerpen
1985	Abitur

Wehrdienst

1985 bis 1986	Bundesluftwaffe

Studium

1986 bis 1994	Studium der Informatik an der RWTH Aachen
1989 bis 1990	Auslandsaufenthalt an der University of Kent at Canterbury / GB
1994	Diplom

Promotion

1994 bis 1998	Wissenschaftlicher Angestellter im Bereich Spracherkennung am Lehrstuhl für Informatik VI der RWTH Aachen
2000	Promotion

Berufstätigkeit

1999 bis heute	Wissenschaftlicher Angestellter im Bereich Mensch–Maschine–Kommunikation an den Philips Forschungslaboratorien Aachen