Karlsruher Institut für Technologie

Automatische Zeichensetzung in Spracherkennungssystemen

Entscheidungsbaum und Sprachmodell im Vergleich

Bachelorarbeit
von

Heike Adel

am Institut für Anthropomatik
der Fakultät für Informatik
Lehrstuhl: Prof. Dr. Alexander Waibel

Erstgutachter:	Prof. Dr. A. Waibel
Zweitgutachter:	Dr. S. Stüker
Betreuender Mitarbeiter:	Dipl.-Inform. K. Kilgour

Bearbeitungszeit: 06. Juni 2011 – 30. September 2011

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 30. September 2011

Abstract

In dieser Arbeit wird die Möglichkeit untersucht, Spracherkennerausgaben in englischer Sprache durch Satzzeichen automatisch zu strukturieren.
Das im Rahmen dieser Bachelorarbeit erstellte System behandelt die Satzzeichen Punkt, Komma und Fragezeichen.
Es werden verschiedene Ansätze diskutiert und zwei stochastische Modelle getestet und bewertet: ein Hidden-N-Gramm-Sprachmodell und ein Entscheidungsbaum.
Das N-Gramm-Sprachmodell arbeitet nur auf den Worten eines Textes, der Entscheidungsbaum bezieht prosodische Merkmale und Wortarten mit ein und dafür kaum Worte. Als prosodische Merkmale werden Pausen nach dem aktuellen Wort, eine Wortlängen-Wortsprechdauerrelation sowie die Information, ob nach dem aktuellen Wort ein Sprechwechsel vorliegt, betrachtet. Die Arbeit zeigt, dass die prosodischen Merkmale des Entscheidungsbaums allein zu ähnlichen Ergebnissen führen können wie die Worte des Sprachmodells. Allerdings wird auch deutlich, dass Entscheidungsbaum und Sprachmodell kombiniert zu deutlich besseren Ergebnissen führen als einzeln.
Die in der Arbeit verwendeten Modelle setzen die Zeichen, die sie finden, zufriedenstellend, allerdings erkennen sie zu wenige Zeichen. Das erstellte Endsystem beinhaltet daher nicht nur die reine Kombination der beiden Modelle sondern auch noch einen Faktor, der dafür sorgt, dass die Wahrscheinlichkeit für „kein Zeichen“ heruntergewichtet und die Wahrscheinlichkeiten für die Satzzeichen entsprechend hochgewichtet werden. Für diesen Faktor werden zwei mögliche Methoden vorgestellt, bei denen die eine einen konstanten Faktor nimmt und die andere die bisherige Wortanzahl seit dem letzten Satzende mit einbezieht.
Beide Methoden führen zu ähnlich guten Ergebnisse. Am besten erweist sich eine Kombination: Das Sprachmodell wird mit dem konstanten Faktor 0,8 umgewichtet, der Entscheidungsbaum abhängig von der bisherigen Wortzahl k nach der Gleichung $1-(0{,}6\cdot k+0{,}4)$. Im Anschluss werden die Wahrscheinlichkeiten der Modelle interpoliert, wobei das Sprachmodell mit Gewicht 0,7 und der Entscheidungsbaum mit Gewicht 0,3 einfließt.
Dieser Ansatz führt zu einer Fehlerrate bei der Satzgrenzenerkennung von 65,95% auf den Hypothesen des Spracherkenners sowie zu einer Fehlerrate von 45,83 % auf den Referenztexten.

Inhaltsverzeichnis

1. Einleitung

Automatische Spracherkennung spielt im täglichen Leben bereits eine große Rolle und wird den Alltag immer weiter durchdringen. So kann man beispielsweise seinen Computer, sein Handy oder auch das Navigationsgerät im Auto durch Spracheingaben steuern. Ob Menschen das Spracherkennungssystem als gut oder schlecht empfinden, orientiert sich daran, ob es die gewünschten Befehle richtig versteht und interpretiert. Wird die Spracherkennung nicht zur Steuerung genutzt, sondern soll der gesprochene Text zum Lesen dargeboten werden, so wird die Akzeptanz durch die Benutzer nicht nur von der Korrektheit der Transkription sondern auch davon abhängen, ob sie die Ausgabe angenehm lesen können. Hierbei spielen Satzzeichen, die einen Wortstrom strukturieren, eine große Rolle.
Diese Arbeit beschäftigt sich mit dem Problem, einen unstrukturierten englischsprachigen Text automatisch mit Satzzeichen zu versehen.
Wie stark die Lesbarkeit eines Textes von seiner Struktur und damit von Satzzeichen abhängen kann, wird im weiteren Verlauf des Einleitungskapitels gezeigt.

1.1 Zielsetzung der Arbeit

Liest man sich die Ausgabe eines Spracherkenners durch, hat man aufgrund der fehlenden Satzzeichen Mühe, einst vorhandene Strukturen wiederzuerkennen.
Das folgende Beispiel verdeutlicht, wie viel leichter es fällt, einen strukturierten Text zu lesen, im Gegensatz zu einem Satz ohne Satzzeichen:

> you see Bob Geldof in that clip that you played earlier on said there was not a single shred of evidence of a diversion of funds and I have to say to you that's not true there is a lot of evidence

> You see, Bob Geldof, in that clip that you played earlier on, said there was not a single shred of evidence of a diversion of funds. And I have to say to you, that's not true, there is a lot of evidence.

Dieser Beispielsatz ist einem Podcast des BBC vom 10.03.2010 entnommen.
Das Beispiel zeigt deutlich, dass beim Sprechen noch vorhandene Satzstrukturen im

Nachhinein in die Transkription wieder eingefügt werden müssen, um die Ausgabe lesbar zu gestalten.
Aber Zeichensetzung führt nicht nur zur besseren Lesbarkeit der Ausgabe sondern auch zum richtigen Verständnis eines Satzes: Sie kann Doppeldeutigkeiten vermeiden.
Als Beispiel sei der folgende kurze Brief genannt. [1] Je nachdem, wo man die Satzzeichen setzt, ergibt sich ein anderer Sinn:

> Dear Jack,
> I want a man who knows what love is all about. You are generous, kind, thoughtful. People who are not like you admit to being useless and inferior. You have ruined me for other men. I yearn for you. I have no feelings whatsoever when we're apart. I can be forever happy - will you let me be yours?
> Jill

> Dear Jack,
> I want a man who knows what love is. All about you are generous, kind, thoughtful people, who are not like you. Admit to being useless and inferior. You have ruined me. For other men I yearn. For you I have no feelings whatsoever. When we're apart I can be forever happy. Will you let me be?
> Yours,
> Jill

Welcher Sinn von der Schreiberin beabsichtigt war, lässt sich nicht erkennen, wenn man den Brief als Reinform ohne Satzzeichen vorliegen hat. Ähnlich kann auch nicht immer eindeutig eine Spracherkennerausgabe im Nachhinein mit Zeichen versehen werden. Dies lässt bereits erahnen, dass es neben textuellen Hinweisen noch weitere Hinweise auf Zeichensetzung geben muss, gerade wenn es sich um gesprochene Texte handelt: Würde Jill ihren Brief vorlesen, würde sie an den Stellen, an denen sie ihre Satzzeichen vorgesehen hat, Pausen einlegen.
Diese Arbeit wird sich mit der Frage beschäftigen, auf welche Arten und mit welchen Ergebnissen die Spracherkennerausgaben mit Hilfe von Punkt, Komma und Fragezeichen wieder strukturiert werden können. Dies verbessert nicht nur das menschliche Verständnis sondern kann auch in der weiteren automatischen Verarbeitung des Textes von Vorteil sein: Maschinelle Übersetzung wird vereinfacht, wenn bereits Satzstrukturen erkannt worden sind. [2] Auch das automatische Erstellen von Zusammenfassungen des vorliegenden Textes wird begünstigt. [3]
Fokus dieser Arbeit liegt allerdings nicht auf der Zeichensetzung in einem als Ganzes vorliegenden Transkript sondern vielmehr auf der Zeichensetzung in Echtzeit. Spricht man in ein Mikrophon, so soll der Text nicht nur online erkannt sondern auch gleich mit Satzzeichen versehen werden. Die erstellten Modelle werden zwar offline mit einem kompletten Transkript bewertet, können auf diese Art aber in den Lecture Translator des KIT integriert werden. [4]

1.2 Gliederung der Arbeit

Im ersten Kapitel „Grundlagen“ werden wichtige Begriffe erklärt, die für das Verständnis der weiteren Ausführungen nötig sein werden. Außerdem wird erläutert, welche Lösungsansätze es bisher in der Literatur gibt und zu welchem Ergebnis sie kommen.
Das zweite Kapitel „Analyse“ beschreibt die Umgebung, in der der erstellte Programmcode ausgeführt wird. In diesem Zusammenhang werden auch die für diese Arbeit in Frage kommenden Lösungsansätze näher diskutiert.
Im „Entwurf“ werden die Modelle dieser Arbeit inklusive ihrer Parameter beleuchtet.
Das vierte Kapitel „Implementierung“ beschreibt die verwendeten Programme zur Realisierung des gewählten Ansatzes.
In der „Evaluierung“ werden die Ergebnisse vorgestellt und analysiert.
Der letzte Teil „Fazit und Ausblick“ fasst die wichtigsten Erkenntnisse zusammen und gibt einen Ausblick auf weitere mögliche Ansätze zur Verbesserung der Ergebnisse.

2. Grundlagen

In diesem Kapitel wird zunächst auf den grundsätzlichen Aufbau eines Spracherkennungssystems eingegangen, da das im Rahmen dieser Arbeit erstellte System auf den Ausgaben eines solchen aufbaut.
Des Weiteren werden für das Verständnis der verwendeten Ansätze wichtige Begriffe erklärt.
Schließlich werden frühere Arbeiten vorgestellt, die für die automatische Zeichensetzung relevant sind.

2.1 Grundlagen eines Spracherkenners

Abbildung 2.1 stellt den grundsätzlichen Ablauf eines Spracherkennungsprozesses dar. Das Audiosignal, das nach der Aufnahme durch ein Mikrophon als analoge elektrische Welle vorliegt, muss für die Verarbeitung durch einen Computer zunächst digitalisiert werden. Dazu wird es abgetastet und quantisiert. Da das menschliche Gehör auf einer Frequenzanalyse basiert und die Hauptinformationen von Sprache im Frequenzbereich eines Audiosignals liegen, wird das Signal nachfolgend durch Fouriertransformation in den Frequenzbereich überführt. Zudem führt man eine Filterung durch und berechnet Cepstralkoeffizienten, die (zusammen mit ihren Beschleunigungskoeffizienten) die Merkmalsvektoren bilden, welche an den Spracherkenner übergeben werden. Da der eigentliche Spracherkennungsprozess erst im nun Folgenden beginnt, wird die Vorverarbeitung des Signals in dieser Arbeit nicht näher erläutert und ist in der Abbildung zu einem Vorverarbeitungsblock zusammengefasst.
Im Spracherkenner wird diejenige Wortsequenz ermittelt, die am besten (das heißt mit der größten Wahrscheinlichkeit) zu den beobachteten Merkmalsvektoren passt. Dies lässt sich durch folgende Formel ausdrücken, wobei W eine mögliche Wortsequenz und X die beobachteten Merkmalsvektoren bezeichnet:

$$\arg\max_{W} P(W|X)$$

Nach Bayes Formel lässt sich die gesuchte Wahrscheinlichkeit wie folgt umstellen:

$$\arg\max_{W} P(W|X) = \arg\max_{W} \frac{P(X|W) * P(W)}{P(X)} = \arg\max_{W} P(X|W) * P(W)$$

Die letzte Gleichheit gilt, da über alle Wörter W maximiert wird und somit $P(X)$ für die Maximierung keine Rolle spielt.
Das Sprachmodell, das auf rein linguistischen Kenntnissen basiert, gibt für jede mögliche Wortsequenz die Wahrscheinlichkeit $P(W)$ an. Diese ist zusammengesetzt aus den Wahrscheinlichkeiten für jedes enthaltene Einzelwort, im gegebenen Kontext beobachtet zu werden: $P(W) = P(w_0) \cdot P(w_1|w_0) \cdot ... \cdot P(w_n|w_0...w_{n-1})$
Im akustischen Modell wird die Wahrscheinlichkeit ermittelt, dass auf Basis einer Wortsequenz die beobachteten Merkmalsvektoren entstehen können. Dies lässt sich ausdrücken durch $P(X|W)$.
Die beschriebene Suche nach den wahrscheinlichsten Worten, das heißt, die Ermittlung des argmax, wird auch Decoding genannt.
Oben genannte Formel des Spracherkenners lässt sich somit durch Sprachmodell und akustisches Modell und Decoder lösen.
In einem Phonem-basierten Spracherkenner, der im akustischen Modell die Wahrscheinlichkeiten nicht auf Basis von Wörtern sondern auf Basis von Phonemen ermittelt, benötigt man zudem noch ein Wörterbuch, das für jedes Wort angibt, aus welchen Phonemen es sich zusammensetzt. Als Phonem bezeichnet man in der Linguistik die kleinste linguistische Einheit, die in einer Sprache zu einem Bedeutungsunterschied führt. Beispielsweise sind im Englischen /h/ und /m/ Phoneme, da die Wörter „house“ und „mouse“ eine andere Bedeutung besitzen.
Die wahrscheinlichste Wortsequenz wird schließlich als Transkription ausgegeben. Ausgabe des Spracherkenners sind somit im Idealfall exakt die Worte, die in dem Audiosignal gesprochen wurden. Da Satzzeichen nicht mitgesprochen werden, werden diese demzufolge auch nicht transkribiert.

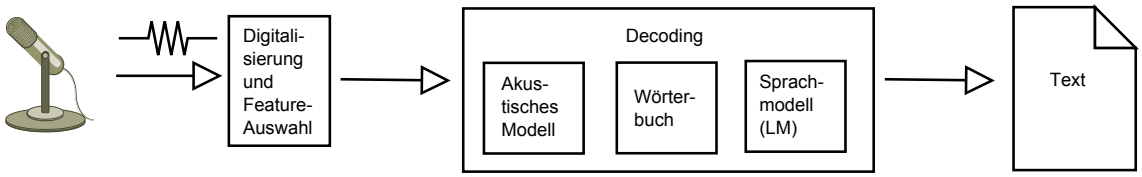

Abbildung 2.1: Der Spracherkennungsprozess
(Abbildung inspiriert von den Vorlesungsfolien zu „Kognitive Systeme“ [5])

2.2 Linguistische Begriffe

Wie bereits in der Einleitung motiviert gibt es nicht nur textuelle Merkmale für die Erkennung von Satzstrukturen. Die folgenden Unterabschnitte erklären Begriffe für die Definition weiterer Elemente von Sprache.
Eine erste wichtige Erkenntnis stellt die Tatsache dar, dass sich gesprochene und geschriebene Phrasen voneinander unterscheiden können.

2.2.1 Sentence-like Units

Für korrekte Zeichensetzung benötigt man Kenntnisse über Satzgrenzen. Hier spielt der Begriff der „Sentence-like Unit“ (SU), zu deutsch „satzähnliche Einheit“, eine große Rolle. Er beschreibt eine zusammengehörige Phrase innerhalb eines Textes.

Im Allgemeinen sind Phrasen in frei gesprochenen Texten kürzer als in gelesenen. [6, 7]
Dies zu wissen ist nötig, um Fehlerraten korrekt interpretieren zu können. Ein Freisprechender wird in der Regel mehr Pausen in seinen Text einbauen als ein Vorlesender, der sich an den im Text vorhandenen Kommata und Punkten orientieren wird. Des Weiteren gibt es in frei gesprochenen Texten auch unvollständige SUs, wenn der Sprecher sich selbst unterbricht oder unterbrochen wird. Eine Möglichkeit, SUs in einem gesprochenen Text zu erkennen, liegt in der Betrachtung der Prosodie des Sprechers. Dies zeigen auch vergangene Arbeiten (vergleiche Abschnitt 2.5).

2.2.2 Prosodie

Unter dem Begriff der Prosodie werden Sprechmerkmale wie Betonung, Sprechrhythmus und Sprechmelodie zusammengefasst. [8] Als Hauptaspekte der Prosodie werden Dauer (von Pausen oder Endlauten eines Wortes), der Verlauf der Sprechfrequenz (und damit der Tonhöhe) und die Energie (und damit die Lautstärke) angesehen. [9]
Ein Sprecher strukturiert mit prosodischen Merkmalen seinen Satz: Mit Pausen beispielsweise trennt er einzelne SUs ab; mit seiner Sprechmelodie zeigt er, ob er gerade eine Aussage, Frage oder einen Ausruf artikuliert.

2.2.3 Part-of-speech

Neben den Eigenschaften, die durch die Aussprache von Wörtern entstehen, können diesen auch auf textueller Ebene Merkmale zugeordnet werden. Jedes Wort gehört einer Wortart an (auf Englisch „Part-of-speech“, kurz POS). Durch Wortarten können Wörter in Klassen eingeordnet und somit verallgemeinert werden.

2.3 Erläuterung wichtiger Modelle

Die folgenden Unterpunkte erklären das grundlegende Prinzip der Methoden, die in vergangenen Arbeiten verwendet beziehungsweise kombiniert wurden.
In den vorgestellten Formeln treten einige Variablen wiederholt auf, weshalb sie nur einmal an dieser Stelle erklärt werden: So steht w für ein Wort, h für die Vorgeschichte des Wortes und X für die akustischen Merkmalsvektoren. Als Event werden Ereignisse bezeichnet, die zu einer Beobachtung gehören, oftmals aber nicht sichtbar sind. Im Falle von Zeichensetzungsmodellen stellen die Zeichen Events dar.
Da in der stochastischen Spracherkennung der Informationsgehalt und damit die Entropie eine große Rolle spielt, wird zunächst ein Modell erklärt, das auf Basis der maximalen Entropie seine Entscheidungen trifft.

2.3.1 Maximum Entropie (MaxEnt)

Die Maximum-Entropie-Methode ist ein grundlegender Ansatz zur Bestimmung der besten Wahrscheinlichkeitsverteilung in der auf Statistik basierenden Spracherkennung. Dabei werden nur bestehende Annahmen betrachtet, aber keine weiteren getroffen. Die bestehenden Annahmen können als Merkmalsfunktionen (Indikatorfunktionen, die den Wert 1 annehmen, wenn die Annahme zutrifft und sonst den Wert 0) modelliert werden. Es wird die Wahrscheinlichkeitsverteilung gewählt, die zur

maximalen Entropie führt. Dazu dient folgende Formel, wobei f_j die j-te Merkmalsfunktion bezeichnet und λ_j das Gewicht, mit dem sie einfließt. Die Anzahl aller vorhandenen Merkmalsfunktionen ist J.

$$P_{me}(w|h) = \frac{e^{\sum_{j}^{J} \lambda_j \cdot f_j(w,h)}}{Z(h)} \text{ [7, 10]}$$

Der Nenner Z(h) dient zur Normalisierung, damit alle Wahrscheinlichkeiten für gegebenes h aufsummiert über alle w den Wert 1 ergeben. Er berechnet sich wie folgt:

$$Z(h) = \sum_{w} e^{\sum_{j}^{J} \lambda_j \cdot f_j(w,h)} \text{ [7, 10]}$$

Das Ziel stellt die Angleichung der erwarteten Merkmale an die empirisch untersuchten dar, das heißt

$$E_P[f_j(w,h); \lambda_j] = E_{\tilde{P}}[f_j(w,h)] \text{ [10]}$$

wobei die rechte Seite der Gleichung als relative Häufigkeit des Merkmals j in den Trainingsdaten bestimmt wird.
Da auch die Zeichensetzung ein Problem darstellt, bei dem man wahrscheinlichkeitsbasiert zwischen den einzelnen Zeichen auswählen kann, lässt sich dort die MaxEnt-Methode anwenden, indem man als w die möglichen Zeichen und als h die Worte oder prosodischen Merkmale betrachtet und Indikatorfunktionen für ihr gemeinsames Auftreten aufstellt. [10] Das Prinzip der maximalen Entropie wird in weiteren Modellen ebenfalls verwendet. So greift auch die Erstellung eines Entscheidungsbaums auf die Entropie zurück.

2.3.2 Entscheidungsbaum

Ein Entscheidungsbaum dient zur Verwertung vorhandener Merkmale. In einem Baum werden hierarchisch Fragen nach diesen Merkmalen angeordnet und abgeprüft. Je nachdem, welchen Wert das getestete Merkmal im konkreten Fall besitzt, das heißt, wie die Frage beantwortet wird, wird ein anderer Weg nach unten gewählt. Die Blätter bilden die am besten passenden Entscheidungen. [11]
Die Reihenfolge der Fragen wird anhand deren Informationsgehalt und damit anhand der Entropie ermittelt.
In der Zeichensetzung beispielsweise kann mit Hilfe eines Entscheidungsbaums anhand von textuellen oder prosodischen Merkmalen entschieden werden, welcher Pfad durch den Baum genommen wird. Auf den Blättern stehen die zu setzenden Zeichen. Abbildung 2.2 zeigt ein Beispiel für den Aufbau eines Entscheidungsbaums. Da ein solcher Baum weit nach unten wachsen kann, kann man mit ihm einen größeren Kontext als mit einem N-Gramm abdecken.

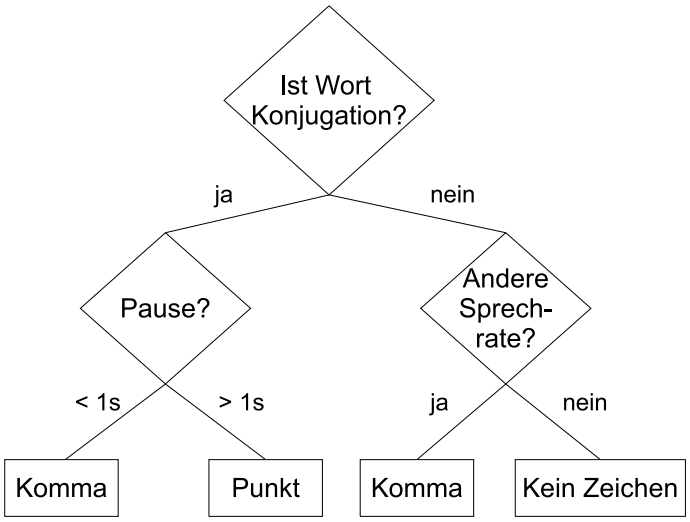

Abbildung 2.2: Aufbau eines Entscheidungsbaums

2.3.3 N-Gramme

Der Fokus von N-Grammen liegt auf der Berechnung der Wahrscheinlichkeit eines Wortes oder Events unter der Bedingung einer gewissen Vorgeschichte. Problematisch dabei erweist sich die Länge der Vorgeschichte: Zum Einen stellt es sich als nicht effizient heraus, die Wahrscheinlichkeit jeder möglichen Historie vorzuberechnen, um dann jeweils das wahrscheinlichste Wort bei gegebener Historie auswählen zu können. Grund dafür ist, dass sich bei langer Historie viel zu viele mögliche Kombinationen aus Historien und Wörtern ergeben. Zum Anderen ließe sich eine große Historie nicht effizient trainieren: Die Wahrscheinlichkeiten werden nämlich wie im Folgenden vorgestellt aus den relativen Auftrittshäufigkeiten der Kombinationen in den Trainingstexten ermittelt. Dort lassen sich aber nicht alle möglichen Kombinationen auffinden, da ein Trainingstext nie alle möglichen Wortfolgen abdecken kann. Als Lösung betrachtet man eine beschränkte Historie der Länge n-1, insgesamt mit dem aktuellen Wort somit n Worte. Hieraus leitet sich der Begriff N-Gramm ab. Die stochastische Grundlage für diese Verkleinerung der Historie bildet die Markov-Annahme (n-1)-ter Ordnung. Sie besagt, dass nur die letzten (n-1) Worte entscheidend für das aktuelle Wort sind. Die Wahrscheinlichkeit für das k-te Wort errechnet sich damit nach folgender Formel:

$$P(w_k|h) = P(w_k|w_{k-(n-1)}w_{k-(n-2)}...w_{k-1})$$

Diese Wahrscheinlichkeit für $P(w|h)$ ergibt sich wie folgt aus den Trainingsdaten:

$$P(w|h) = \frac{Count(hw)}{Count(h)}$$

Da N-Gramme in heutigen Spracherkennern standardmäßig als Sprachmodelle eingesetzt werden, werden die beiden Begriffe in dieser Arbeit äquivalent verwendet.

Hidden N-Gramme

Bei Versteckten N-Grammen (englisch: hidden n-grams) gibt es zusätzlich zum Vokabular bestehend aus allen Wörtern des Spracherkenners noch ein Vokabular aus

versteckten Wörtern. Diese können zum Beispiel Satzzeichen sein. Bei der Berechnung der Wahrscheinlichkeiten werden diese ebenfalls beachtet. Ein Text ohne Satzzeichen kann dann aufgrund der ermittelten Wahrscheinlichkeiten für die Satzzeichen unter der Bedingung der vorhandenen Wörter mit diesen versehen werden. Dazu wird es als Hidden Markov Modell genutzt, das im nächsten Abschnitt beschrieben wird. Die Wort-Zeichen-Paare aus den Trainingsdaten dienen als Zustände, die Worte als Beobachtungen. Als Übergangswahrscheinlichkeiten werden die N-Gramm-Wahrscheinlichkeiten verwendet. [9]

2.3.4 Hidden Markov Modelle (HMM)

Versteckte Markov-Modelle (Hidden Markov Models, HMMs) bestehen aus Zuständen, die mit gewissen Wahrscheinlichkeiten Beobachtungen emittieren. Das System kann dabei von einem Zustand in andere Zustände übergehen. Formal ist ein HMM damit ein Fünftupel bestehend aus Zuständen, Übergangswahrscheinlichkeiten, Emissionswahrscheinlichkeiten, einem Vokabular, aus dem die Beobachtungen stammen, sowie Initialwahrscheinlichkeiten, die für jeden Zustand angeben, wie wahrscheinlich die Berechnung dort beginnt.
Der Begriff „hidden“ bedeutet, dass man nicht weiß, welche Zustände durchlaufen werden: Man kennt nur die Beobachtungen und errechnet anhand der Übergangs- und Emissionswahrscheinlichkeiten die am wahrscheinlichsten durchlaufene Zustandsfolge. Das Modell basiert auf der „Markov-Annahme“, die besagt, dass für die Wahrscheinlichkeit des aktuellen Zustands nur ein Zustand davor betrachtet werden muss, nicht aber die insgesamt durchlaufene Sequenz:

$$P(q_t = j | q_{t-1} = i, q_{t-2} = k...) = P(q_t = j | q_{t-1} = i)$$

q_t: Zustand, in dem sich das System im Zeitpunkt t befindet;
i, j, k: Zustände des Modells

Hidden Markov Modelle werden in der Spracherkennung genutzt, um Wörter zu erkennen: Man errechnet, welche Phoneme beziehungsweise Wörter (Zustände) am wahrscheinlichsten zu den vorhandenen Merkmalsvektoren (Beobachtungen) geführt haben. Analog kann man für die Zeichensetzung Wort-Zeichen-Paare als Zustände modellieren und prosodische Merkmale und Wörter als Beobachtungen. Abbildung 2.3 zeigt dazu ein Beispiel.

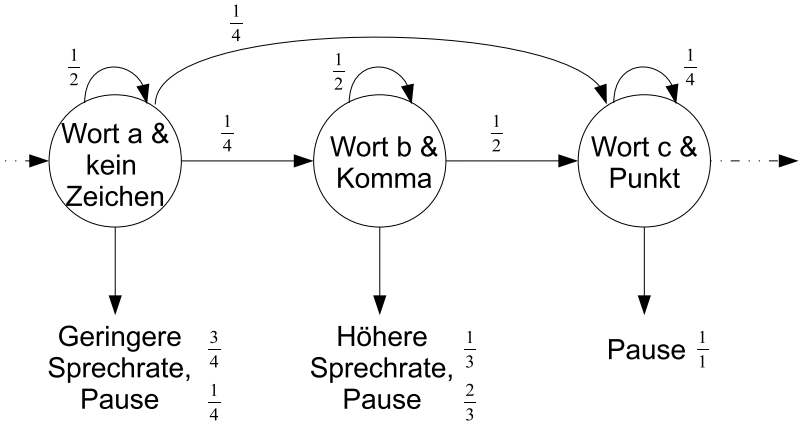

Abbildung 2.3: Aufbau eines HMMs

Nachteile von HMMs stellen die starken Unabhängigkeitsannahmen dar. So sind die Emissionswahrscheinlichkeiten immer genau einem Zustand zugeordnet. Damit könnte man nicht modellieren, dass eine Beobachtung beispielsweise von einer bestimmten Folge aus mehreren Zuständen abhängt. Dieses Problem beheben Conditional Random Field-Modelle.

2.3.5 Conditional Random Fields-Modelle (CRF)

Conditional Random Fields dienen der Bearbeitung sequenzieller Daten, beispielsweise einer Wortsequenz. Sie können diese je nach Bedarf mit Tags versehen, segmentieren oder ähnliches. [12]
Es handelt sich um ungerichtete Graphen, die eine Eventsequenz Y repräsentieren, die von einer globalen Beobachtungssequenz X abhängt. [12–14] Die Ungerichtetheit stellt einen ersten Unterschied zu HMMs dar.
Im Falle der Zeichensetzung werden mit X die Worte beziehungsweise Prosodie modelliert und mit Y die zu setzenden Zeichen.
Im Gegensatz zum Hidden Markov Modell wird beim Conditional Random Fields-Modell nicht die gemeinsame Verteilung von Beobachtung und zugrundeliegendem Event berechnet sondern nur die Wahrscheinlichkeit des zugrundeliegenden Events bei gegebener Beobachtung. Dadurch sind die Unabhängigkeitsannahmen weniger streng und die Beobachtung, die zu Testzeiten sowieso fest gegeben ist, muss nicht mit modelliert werden. [15] Die bedingten Wahrscheinlichkeiten können durch Merkmalsfunktionen bestimmt werden, die beliebig große Abhängigkeiten einbeziehen können. [12] CRF-Modelle können damit die gesamte Beobachtungssequenz verwenden, während HMMs nur die lokal aktuellen Beobachtungen sehen.

Linear Chain CRF

Die einfachste Graphrepräsentation ist die einer linearen Kette: Die Eventsequenz wird als Kette durchlaufen. Damit ergibt sich folgende Wahrscheinlichkeitsverteilung:

$$P_\lambda(Y|X) = \frac{1}{Z(X)} \cdot e^{\sum_k \lambda_k \cdot F_k(Y,X)} \quad [7,\ 13,\ 14]$$

F_k sind dabei entweder Merkmalsfunktionen des Events und der Beobachtung zu einem Zeitpunkt oder Zustandsübergangsmerkmalsfunktionen. k stellt die verschie-

denen Merkmale dar. Z ist ebenso wie bei den Maximum-Entropie-Modellen ein Normalisierungsfaktor. Er hat folgende Form:

$$Z_\lambda(X) = \sum_Y e^{\sum_k \lambda_k \cdot F_k(Y,X)} \quad [7]$$

Die durchlaufene Kette kann man sich zum Beispiel wie in Abbildung 2.4 gezeigt vorstellen.

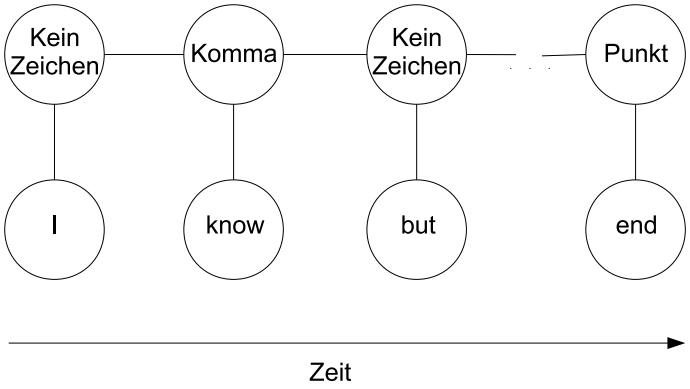

Abbildung 2.4: Linear Chain CRF
(Grafik inspiriert von [12, 13, 15])

Factorial CRF

Bei Factorial Conditional Random Fields werden mehrere Events betrachtet, die auch wieder voneinander abhängen können. [15] Dies könnte beispielsweise eine wie in Abbildung 2.5 dargestellte Form haben.

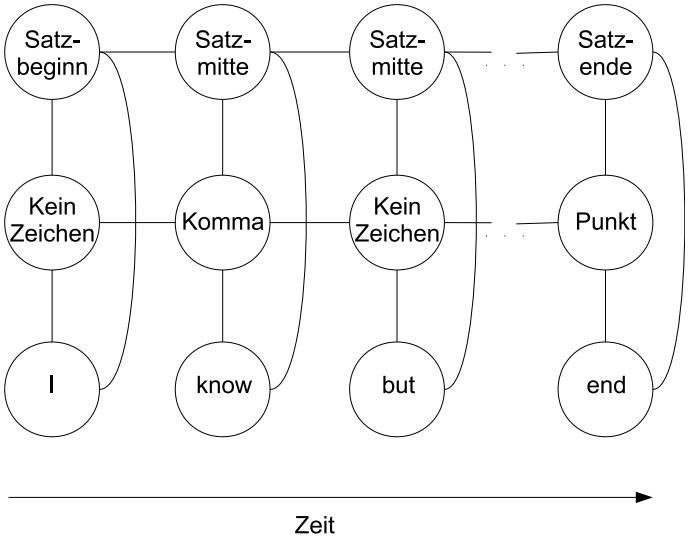

Abbildung 2.5: Factorial CRF
(Grafik inspiriert von [15])

2.4 Evaluationsgrundlagen

Dieser Abschnitt stellt die in der Spracherkennung üblichen Bewertungsmaßstäbe vor und wendet sie auf das Zeichensetzungsproblem an.
Im Folgenden werden die korrekten Satzzeichen aufgrund der englischen Bezeichnung corrects mit C abgekürzt, die fälschlicherweise gesetzten mit I (von insertions), ausgelassene mit D (von deletions) und verwechselte mit S (von substitutions).
Als Bewertungsmaße werden Precision, Recall, Slot-Errorrate und Sentence-Unit-(SU)-Errorrate verwendet.

2.4.1 Precision

Precision misst die Genauigkeit der Hypothese, das heißt, wie viele der erkannten Zeichen auch korrekt sind.
Dies ergibt sich durch die Formel:

$$P = \frac{C}{C + S + I} \text{ [15–18]}$$

Der Nenner dieser Formel entspricht allen Satzzeichen in der Hypothese.
Das Precision-Maß wird in dieser Arbeit nicht nur für die konkreten Zeichen berechnet sondern auch für die erkannten Satzgrenzen, da der Fokus eher auf der generellen Strukturierung des Textes durch Zeichen liegt. Analog zu vergangenen Arbeiten wird dafür eine SU-Precision betrachtet, die die Präzision der Strukturierung bewertet. [13] Dies lässt sich durch folgende Formel ausdrücken:

$$P_{SU} = \frac{C + S}{C + S + I}$$

2.4.2 Recall

Recall misst die Trefferquote des Systems, das heißt, wie viele der tatsächlichen Zeichen auch korrekt erkannt wurden.
Dies ergibt sich durch die Formel:

$$R = \frac{C}{C + S + D} \text{ [15–18]}$$

Der Nenner dieser Formel entspricht allen Zeichen im Referenztext.
Ebenso wie das Precision-Maß wird auch das Recall-Maß nicht nur für die Zeichen sondern auch für die Strukturierung als Ganzes berechnet. Dazu wird der SU-Recall betrachtet, der die Trefferquote der grundsätzlichen Strukturierung bewertet [13] und sich dabei analog zur SU-Precision wie folgt definieren lässt:

$$R_{SU} = \frac{C + S}{C + S + D}$$

2.4.3 Slot-Errorrate (SER)

Während Precision und Recall zwei unabhängige Maße darstellen, um ein Klassifikationssystem zu bewerten, bietet die Slot-Error-Rate eine Möglichkeit zur Kombination. In dieser Arbeit wird zur Kombination nicht das oft verwendete F-Maß benutzt, das sich durch folgende Formel berechnet:

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \text{ [15–18]}$$

Grund dafür sind die Ausführungen von John Makhoul [18], der beschreibt, dass das F-Maß Deletion- und Insertion-Fehler abwertet. Als Alternative stellt er die Slot-Errorrate vor, die auch in dieser Arbeit verwendet wird. Sie setzt alle Fehler des Systems mit allen Zeichen im Referenztext in Beziehung:

$$SER = \frac{S + D + I}{C + S + D} \text{ [17, 18]}$$

2.4.4 SU-Errorrate

Das Maß SU-Errorrate ist den anderen Arbeiten auf dem Gebiet der automatischen Zeichensetzung entnommen. Es wird nach folgender Formel berechnet:

$$SU - Error = \frac{I + D}{C + S + D} \text{ [7, 19]}$$

Die falsch erkannten Zeichen werden mit allen Zeichen des Referenztextes in Beziehung gesetzt. Das Maß ähnelt dabei der Slot-Errorrate, jedoch werden Zeichenverwechslungen nicht als Fehler angesehen. Dies erweist sich ebenso wie SU-Precision und SU-Recall als sinnvoll, da der Fokus der Arbeit auf der Strukturierung von Wortfolgen liegt. Dazu ist es besser, ein falsches Zeichen zu erkennen als nichts einzufügen.
Während die Precision- und Recall-Werte für ein gutes System möglichst hoch sein sollten, erweist sich eine niedrigere SER beziehungsweise SU-Fehlerrate als besser.

2.5 Verwandte Arbeiten

Dieser Teil beschreibt Arbeiten, die auf dem Gebiet der automatischen Zeichensetzung veröffentlicht wurden. Die folgenden Unterkapitel zeigen verwendete Ansätze und ihre Ergebnisse auf. Sie lassen sich unterteilen in rein textbasierte Ansätze und Ansätze, die textuelle und prosodische Merkmale kombinieren.

2.5.1 Rein textbasierte Ansätze

Zunächst erfolgt eine Vorstellung der textbasierten Ansätze. Sie betrachten nur Worte als Hinweise für die Zeichensetzung.

2.5.1.1 N-Gramm-Sprachmodell

Gravano [20] trainiert ein N-Gramm-Sprachmodell und implementiert für jede mögliche Wortsequenz einen Automaten. Dieser enthält als Zustände die vorkommenden Wörter und an den Zustandsübergängen die möglichen Zeichensetzungssymbole. Die Wahrscheinlichkeiten für die Zustandsübergänge ergeben sich aus dem Sprachmodell. Es wird jeweils der Weg durch den Automaten mit den höchsten Wahrscheinlichkeiten gewählt.
Diese Vorgehensweise führt zu folgenden Ergebnissen:

- bei Kommata: Precision von 50,37% (bei 3-grams) und Recall von 59,57% (bei 6-grams)
- bei Punkten: Precision von 55,52% (bei 6-grams) und Recall von 65,53% (bei 6-grams)

- bei Fragezeichen: Precision von 49,17% (bei 6-grams) und Recall von 35,11% (bei 6-grams)

Gravano zeigt, dass die N-Gramm-Ordnung kaum eine Rolle spielt, die Größe des Trainingssets sich dagegen als sehr wichtig erweist: Je größer das Trainingsset desto bessere Ergebnisse erhielt er. Dass Fragezeichen nicht so gut erkannt werden wie Kommata oder Punkte, liegt an dem beschränkten Kontext von N-Grammen: Das Fragewort steht in vielen Fällen mehr als n Wörter vom Fragezeichen entfernt und kann daher nicht mehr mit einem N-Gramm-Sprachmodell modelliert werden.
Conditional Random Field-Modelle können dieses Manko beheben. [15]

2.5.1.2 Conditional Random Field-Modelle

Lu [15] betrachtet ebenfalls nur textuelle Merkmale. Für jedes Wort werden Unigram-, Bigram- und Trigramwahrscheinlichkeiten berechnet. In einem ersten Versuch wird ein Linear Chain CRF-Modell verwendet, bei dem jedem Wort als Merkmal das nachfolgende Satzzeichen zugeordnet wird. In einem zweiten Versuch werden zweischichtige Factorial CRFs betrachtet, die nicht nur die Satzzeichen als Tags sondern auch noch die Satzart als Tags definieren (zum Beispiel Aussagesatz oder Fragesatz). So können Fragezeichen von Punkten unterschieden werden, wenn die Frage lang ist. Seine besten Ergebnisse erhält Lu im zweiten Versuch. Diese sind:

- Ergebnisse für den Basic Travel Expression Corpus (enthält touristentypische Sätze):
  Precision von 92,76% und Recall von 84,73%
- Ergebnisse für die Challenge-Task-Datenmenge (enthält Dialoge in der Reisedomäne):
  Precision von 86,69% und Recall von 79,62%

Morgan [13] verfolgt ebenfalls den CRF-Ansatz auf textuellen Merkmalen. Als Event wird ein boolescher Wert benutzt, der angibt, ob ein Wort an einem SU-Anfang auftritt oder nicht. In weiteren Tests wird das System um Part-of-speech-Tags sowie um ein bigram-Sprachmodell ergänzt. Dieses schätzt für jedes Wort die Wahrscheinlichkeit, dass es sich an einem SU-Beginn befindet unter der Bedingung des nachfolgenden Wortes.

- Ergebnisse des Basisversuchs:
  Precision von 79,1% und Recall von 64,49%
- Ergebnisse der ersten Erweiterung (auf POS-Tags):
  Precision von 78,96% und Recall: 65,45%
- Ergebnisse der zweiten Erweiterung (um das Sprachmodell):
  Precision von 78,74% und Recall von 65,76%

2.5.2 Kombination von textuellen und prosodischen Merkmalen

Die folgenden Ansätze kombinieren textuelle und prosodische Merkmale.

2.5.2.1 Maximum-Entropie-Tagger

Jedes Wort bekommt einen Tag zugewiesen (Komma, Punkt, Fragezeichen oder default-Tag). Es werden Merkmale definiert, die prosodische und lexikalische Merkmale kombinieren: Ein Merkmal ist ein Kontext-Tag-Paar, wobei der Kontext durch die beiden vorherigen Worte und deren Tags sowie die beiden Folgeworte bestimmt wird. Des Weiteren werden Pausenkontexte einbezogen. Die Wahrscheinlichkeit eines Tags für das aktuelle Wort wird unter der Bedingung seines Kontextes bestimmt. Die Merkmale und ihre Wahrscheinlichkeiten werden durch Bestimmen jedes Kontext-Tag-Paares im Trainingsset gewonnen. Der Ansatz erreicht folgende Ergebnisse [16]:

- bei Kommata: Precision von 87% und Recall von 73%
- bei Punkten: Precision von 79% und Recall von 85%
- bei Fragezeichen: Precision von 65% und Recall von 27%

2.5.2.2 Mathematische und heuristische Kombination

Im Gegensatz zum vorgestellten Maximum-Entropie-Tagger werden textuelle und prosodische Merkmale in diesem Ansatz nicht gemeinsam betrachtet. Vielmehr werden separat textbasierte und prosodiebasierte Wahrscheinlichkeiten bestimmt und durch zwei Arten kombiniert: zum Einen durch einen mathematischen und zum Anderen durch einen heuristischen Ansatz. [17, 19]
Für die Zeichensetzung wird ein statistisches endliches Zustandsmodell entwickelt, wobei jeder Zustand das gemeinsame Auftreten eines Wortes, eines prosodischen Merkmals und eines Satzzeichens repräsentiert. Die Wahrscheinlichkeit für das Auftreten einer Sequenz der Länge k wird berechnet als Produkt der einzelnen Sequenzinhalte unter der Bedingung der direkt davor auftretenden Worte und Satzzeichen. Die prosodischen Merkmale werden somit als unabhängig voneinander betrachtet, die linguistischen Merkmale mit einem Kontext der Länge 1 in die Vergangenheit. Die Wahrscheinlichkeiten der beinhalteten linguistischen Komponente und die der prosodischen Komponente werden wie erwähnt unabhängig voneinander gewonnen, müssen für das Zustandsmodell aber kombiniert werden.
Im mathematischen Ansatz erfolgt die Kombination nach der Formel:

$$p(\mathrm{s_i}, \mathrm{w_i}, \mathrm{c_i} | \mathrm{w_{i-1}}, \mathrm{c_{i-1}}) \sim \{p^{[P]}(s_i|w_i, c_i)\}^{\alpha} \cdot \{p^{[L]}(w_i, c_i|w_{i-1}, c_{i-1})\} \tag{2.1}$$

Im heuristischen Ansatz erfolgt die Kombination nach der Formel:

$$p(\mathrm{s_i}, \mathrm{w_i}, \mathrm{c_i} | \mathrm{w_{i-1}}, \mathrm{c_{i-1}}) \sim \{p^{[P]}(w_i, c_i|s_i)\}^{\alpha} \cdot \{p^{[L]}(w_i, c_i|w_{i-1}, c_{i-1})\} \tag{2.2}$$

Die besten Ergebnisse liefert das Zustandsmodell, wenn bei der Prosodie nur Pausen betrachtet werden.
Ergebnisse bei Christensen beim mathematischen Ansatz (2.1) sind: [17]

- bei Kommata: Precision von 48% und Recall von 17%
- bei Punkten: Precision von 84% und Recall von 44%

Ergebnisse bei Christensen beim heuristischen Ansatz (2.2) sind: [17]

- bei Kommata: Precision von 54% und Recall von 11%
- bei Punkten: Precision von 79% und Recall von 21%

Gotoh verwendet denselben Ansatz, unterscheidet aber nicht, welches Satzzeichen vorliegt. Er zeigt außerdem, dass das Pausenmodell einzeln betrachtet besser abschneidet als das linguistische Modell allein.
Die besten Ergebnisse erhält er allerdings mit den Kombinationen: [19]

- Prosodisches Modell: Precision von 56% und Recall von 39%
- Linguistisches Modell: Precision von 74% und Recall von 58%
- Mathematischer Kombinationsansatz (2.1): Precision von 66% und Recall von 71%.
- Heuristischer Kombinationsansatz (2.2): Precision von 80% und Recall von 62%

2.5.2.3 Entscheidungsbäume und N-Gramme kombiniert durch HMMs:

Shriberg [21] und Baron [22] nutzen Entscheidungsbäume, um die Wahrscheinlichkeiten von Satzbegrenzungen in Abhängigkeit prosodischer Merkmale zu schätzen. Verwendete Merkmale sind Pausenlängen, Lautlängen (da Menschen gegen Ende eines Satzes langsamer sprechen), Verläufe der Sprechfrequenz und Stimmqualitätsinformationen. Als Kontext werden 200ms sowohl in Vor- als auch in Rückrichtung genutzt. Des Weiteren werden Merkmale wie Sprecherwechsel und Geschlecht hinzugezogen.
Um die textuellen Merkmale, das heißt die gemeinsame Verteilung von Satzgrenzen und Worten (basierend auf 4-grams) zu modellieren, werden HMMs genutzt. Die versteckten Zustände stellen hierbei die Satzzeichen dar. Die Übergangswahrscheinlichkeiten für das HMM ergeben sich aus den 4-gram-Wahrscheinlichkeiten bezogen auf das Trainingsset.
Kombiniert werden die Wahrscheinlichkeiten des Baums und des HMMs auf zwei Arten: Zum Einen werden die Wahrscheinlichkeiten des HMMs und die des Entscheidungsbaums interpoliert. Zum Anderen wird das HMM so erweitert, dass es sowohl Worte als auch prosodische Merkmale emittiert.
Dabei ergeben sich folgende Fehlerraten bezüglich der SU-Grenzen-Erkennung:
Bei Broadcasts-News auf Referenztexten:

- HMM alleine: 4,1%
- Entscheidungsbaum alleine: 3,6%
- Kombination durch Interpolation: 3,5%
- Kombination durch Erweiterung des HMMs: 3,3%

Bei Broadcasts-News auf vom Spracherkenner erkannten Worten:

- HMM alleine: 11,8%

- Entscheidungsbaum alleine: 10,9%
- Kombination durch Interpolation: 10,8%
- Kombination durch Erweiterung des HMMs: 11,7%

Kim [6] trainiert einen Entscheidungsbaum mit Wortarten und prosodischen Merkmalen sowie ein hidden-event-Sprachmodell mit Worten. Als prosodische Merkmale werden Lautlängen, Sprechfrequenzen, Energiestatistiken, Pausen und Sprecherwechsel verwendet. Es werden drei Methoden getestet, um die beiden erhaltenen Wahrscheinlichkeiten zu kombinieren: Bei der ersten Methode (join tree-based modeling) werden die Wahrscheinlichkeiten aus dem Sprachmodell zusammen mit den anderen Merkmalen in einem Entscheidungsbaum modelliert. Die zweite Methode interpoliert Entscheidungsbaum und Sprachmodell linear. Beim dritte Ansatz werden Entscheidungsbaummerkmale, Wörter und Zeichen zusammen mit einem integrierten HMM modelliert. Sowohl Worte als auch Entscheidungsbaummerkmale werden als HMM-Zustandsausgaben betrachtet.
Daraus resultieren folgende Ergebnisse:

- Erste Methode: Fehlerrate: 46,6%
- Zweite Methode: Fehlerrate: 48,4%
- Dritte Methode: Fehlerrate: 48,5%

2.5.2.4 MaxEnt-Modelle im Vergleich mit Hidden-Markov-Modellen und Conditional-Random-Field-Modellen

Liu [7] vergleicht MaxEnt-Modelle, Conditional-Random-Field-Modell und einen HMM-Ansatz. Für die MaxEnt-Modelle wählt er für Wörter, prosodische Merkmale und zugehörige Zeichen Indikatorfunktionen. Bei den Conditional-Random-Field-Modellen verwendet er ebenfalls Indikatorfunktionen, wobei sich die Merkmale dort auf eine Zeichensequenz und nicht nur auf ein einzelnes Zeichen beziehen. Bei dem HMM schätzt er die benötigten Wahrscheinlichkeiten mit einem Entscheidungsbaum. Er erhält folgende Fehlerraten für Spracherkennerausgaben:
Bei Broadcast News:

- HMM-Ansatz: Fehlerrate: 60,64%
- MaxEnt-Modell: Fehlerrate: 58,6%
- CRF-Modelle: Fehlerrate: 48,21%
- Kombination aller drei Ansätze: Fehlerrate: 57,23%

Bei CTS (conversational telefon speech):

- HMM-Ansatz: Fehlerrate: 37,47%
- MaxEnt-Modell: Fehlerrate: 37,39%
- CRF-Ansatz: Fehlerrate: 37,25%
- Kombination aller drei Ansätze: Fehlerrate: 36,26%

3. Analyse

In diesem Kapitel werden zunächst die genauen Anforderungen an das Zeichensetzungssystem beschrieben. Danach folgt eine Bewertung der bereits existierenden Ansätze und die Beschreibung des eigenen Ansatzes.

3.1 Anforderungen

Ziel dieser Arbeit stellt die Strukturierung des von einem Spracherkenner erkannten Wortflusses durch Zeichensetzung dar. Das Setzen der Zeichen wird als Klassifizierungsproblem betrachtet: Je nachdem, ob einem Wort ein Komma, Satzendezeichen oder kein Zeichen folgt, wird es in eine der Klassen „NONE“, „COMMA“ oder „END“ eingeordnet.
Um das System effektiv einsetzen zu können, wird es in das Lecture Translation System des KIT eingebaut. Dieses setzt sich aus dem Spracherkennungssystem JANUS Recognition Toolkit (JRTk) und einer Übersetzungskomponente zusammen. Es kann Meetings in Echtzeit transkribieren und übersetzen. Ziel stellt eine direkte Anzeige des gerade gesprochenen Textes inklusive Satzzeichen dar.
Da das Erkennungssystem das Gesprochene nicht nur erkennt sondern gleichzeitig auch in ausgewählte Sprachen übersetzt, wird das System in Zukunft auch dahingehend eingesetzt werden können, die maschinelle Übersetzung zu verbessern. Diese kann nämlich auf der Struktur durch die Satzzeichen aufbauen und so eventuelle Übersetzungsfehler vermeiden.
Randbedingung des Systems ist frei gesprochene und nicht abgelesene Sprache, die in Echtzeit verarbeitet werden soll.
Diese Arbeit beschränkt sich auf die Zeichen Punkt, Komma und Fragezeichen. Alle anderen Zeichen werden nicht etwa durch Platzhalter ersetzt sondern weggelassen.

3.2 Existierende Lösungsansätze

Folgende Erkenntnisse aus vergangenen Arbeiten erweisen sich für das zu erstellende System als bedeutend:

3.2.1 Erkenntnisse bezüglich Online-Zeichensetzung

Die meisten Ansätze, transkribierte Texte mit Zeichensetzung zu strukturieren, befassen sich mit dem Fall, dass bereits der ganze Text bekannt ist und damit als Kontext auch Worte berücksichtigt werden können, die nach der zu untersuchenden Stelle kommen.
Online-Erkennung, das heißt Erkennung während des Sprechens, erschwert das korrekte Einfügen von Satzzeichen. Grund dafür ist der geringere Kontext für ein Wort. In einem Versuch simulieren Baron [22] und Shriberg [23] Online-Zeichensetzung, indem sie den Kontext nur auf die Vergangenheit beschränken. Dadurch erhöhen sich die Fehlerraten. Im Unterschied zu diesem Versuch, in dem gar kein rechtsseitiger Kontext verwendet wird, wird beim Einbau in den Lecture-Translator jedoch als Kontext der Block verwendet, den der Spracherkenner auf einmal verarbeitet und ausgibt. Gerade zu Beginn des Blocks ist demnach auch rechtsseitiger Kontext vorhanden.

3.2.2 Erkenntnisse bezüglich frei gesprochenen Texten

Frei gesprochene Sprache erschwert die Spracherkennung, da die Sprache Unregelmäßigkeiten, Denkpausen und Selbstverbesserungen aufweisen wird. [6, 7, 16, 19].
Stolcke und Shriberg testen ihren prosodischen Entscheidungsbaum und ihr textuelles Sprachmodell zum Einen auf Referenztexten und zum Anderen auf den Hypothesentexten eines Spracherkenners. [9] Sie stellen fest, dass Spracherkennerfehler zu erhöhten Fehlerraten bei der Satzgrenzenerkennung führen. Als Grund bei dem Sprachmodell geben sie an, dass falsch erkannte Wörter zu Folgefehlern bei der Segmentierung des Wortstroms in Sätze führen. Falsch erkannte Wörter führen zwar zu keinen Problemen beim Entscheidungsbaum, allerdings basieren dessen Eingaben auf den Hypothesen des Spracherkenners, wie der Audiostrom in Wörter zu zerlegen ist. Somit wird auch er von Fehlern des Spracherkenners beeinflusst. Stolcke und Shriberg erreichen auf Referenztexten mit dem Sprachmodell Erkennungsraten von 92,7% und mit dem Entscheidungsbaum Erkennungsraten von 88,9%. Im Falle von Hypothesentexten sinken diese Raten auf 77% beim Sprachmodell und 76,1% beim Entscheidungsbaum.
Ein Experiment von Liu [7] bestätigt dies. Allerdings zeigt sich dort außerdem, dass innerhalb der gesprochenen Texte ebenfalls unterschiedliche SU-Erkennungsraten erreicht werden können: Bei vollkommen frei gesprochener Sprache wie Telefongespräche ergeben sich SU-Fehlerraten von 26,43% (Referenztext) beziehungsweise 36,26% (Hypothesentext), während Broadcasts höhere Fehlerraten von 48,21% (Referenztext) beziehungsweise 57,23% (Hypothesentext) liefern. Grund hierfür könnten längere und damit eindeutigere Sprech- und Denkpausen an Satzgrenzen bei Telefongesprächen sein.

3.2.3 Erkenntnisse bezüglich Prosodie und Text

Gerade bei Spracherkennung, die fehlerhaft ist, erweist sich für die korrekte Erkennung von Satzgrenzen neben der Betrachtung textueller Merkmale auch die Betrach-

tung prosodischer Merkmale als wichtig. Die besten Ergebnisse liefert die Kombination aus beiden. (vergleiche [6, 7, 9, 16, 19, 21, 22])

3.3 Ansatz dieser Arbeit

In diesem Abschnitt wird zunächst die Wahl der Modelle begründet und schließlich dargelegt, welche Merkmale diesen als Eingabe dienen.

3.3.1 Zur Wahl der Modelle

Für diese Arbeit in Frage kommende Alternativen sind MaxEnt-Modelle, Entscheidungsbäume, Hidden-Markov-Modelle und Conditional Random Fields.
Entscheidungsbäume sind für den Menschen interpretierbarer als MaxEnt-Modelle, weshalb ihnen hier den Vorzug gegeben wird. Allerdings basieren auch die Berechnungen bei der Erstellung des Entscheidungsbaums auf Entropiebetrachtungen, wodurch der Entropie-Ansatz indirekt ebenfalls verwendet wird. Im Anwendungsgebiet dieser Arbeit kann nicht immer damit gerechnet werden, dass alle nötigen Kontexte gegeben sind. Modelliert man beispielsweise einen Kontext mit den vergangenen zwei Worten und ein Sprecher beginnt gerade zu reden, so gibt es keine vorherigen zwei Worte. Ebenso kann nicht an allen Stellen der Eingabe mit einem Kontext in die Zukunft gerechnet werden, da es sich um Online-Spracherkennung handelt. Für diese Fälle eignet sich ein Entscheidungsbaum sehr gut, da er es erlaubt, Werte unbesetzt zu lassen. Diese werden bei der Entscheidung nicht etwa irgendwie besetzt. Stattdessen wird der Baum in einem solchen Fall in mehrere Zweige weiterverfolgt.
Eine Alternative zum Entscheidungsbaum wären Conditional Random Field-Modelle, die den Vorteil besitzen, eine gesamte Sequenz zu optimieren und nicht an jedem Punkt, in diesem Fall bei jedem Wort, eine eigene, neue Entscheidung zu treffen. Da sie aber nicht so intuitiv interpretierbar sind wie Entscheidungsbäume, werden sie in dieser Arbeit nicht betrachtet.
Eine weitere Alternative wären Hidden-Markov-Modelle. Ihr großer Nachteil allerdings ist, dass sie aufgrund der Markov-Annahme keine Abhängigkeiten modellieren können, die über einen Vorgängerzustand hinausgehen. Sie sind daher nicht ohne Weiteres auf einen erweiterten Kontext von Satzzeichen anwendbar.
Ein Ansatz, der direkt aus der Sprachmodellierung in der Spracherkennung abgeleitet werden kann, dient in dieser Arbeit als Baseline: Die Erkennung von Satzzeichen mit Hilfe eines Hidden-N-Gramm-Sprachmodells. Er wurde gewählt, weil er auch im Spracherkenner verwendet wird und somit intuitiv als Baseline für weitere Modelle dienen kann.
Eine genauere Beschreibung der eingesetzten Modelle folgt in den Kapiteln 4 und 5.
Die vorangegangenen Arbeiten zeigen, dass sich in den meisten Fällen eine Kombination mehrerer Modelle als sinnvoller erweist als die Modelle einzeln zu verwenden. So werden auch in dieser Arbeit zwei verschiedene Modelle trainiert und schließlich kombiniert. Genaueres zur Kombination folgt ebenfalls in den Kapiteln 4 und 5.

3.3.2 Zur Wahl der Merkmale

Aufgrund der im vorigen Abschnitt genannten früheren Erkenntnisse (3.2, Seite 20) werden auch in dieser Arbeit sowohl textuelle als auch prosodische Merkmale genutzt. Welche prosodischen Merkmale genau verwendet werden, wird im Kapitel 4 ausführlich beleuchtet.

3.4 Zusammenfassung

Der Lecture Translator wird mit einem System erweitert, das es erlaubt, beim Sprechen vorhandene Strukturen wieder in den Textfluss zu bringen. Das System basiert auf prosodischen und textuellen Merkmalen und trifft seine Entscheidung mit Hilfe eines Entscheidungsbaums sowie mit Hilfe eines Sprachmodells.

4. Entwurf

In diesem Kapitel wird der Lösungsansatz genauer beleuchtet und Entwurfsentscheidungen werden dargelegt.

4.1 Genaueres zur Klassifizierung

Die Wörter werden (wie bereits in 3.1 auf Seite 19 beschrieben) in die Klassen „NONE“, „COMMA“ und „END“ eingeteilt. Im Unterschied zu den vorangegangenen Arbeiten wird somit nicht direkt vom System in Punkt und Fragezeichen unterschieden. Grund dafür stellt die Erkenntnis der vorherigen Arbeiten dar, dass Fragezeichen aufgrund des oftmals zu geringen Kontextes nicht sicher erkannt werden können. In diesem System wird daher zunächst nur entschieden, ob der Satz endet oder nicht. Im Falle eines Satzendes wird darauf aufbauend bestimmt, ob der Satz mit einem Punkt oder Fragezeichen endet. Diese Entscheidung wird anhand vorher gesehener Fragewörter beziehungsweise Satzanfängen mit einem Verb getroffen. Die Bestimmung eines Fragezeichens wird dadurch eine Ebene höher geschoben. Eine Liste der verwendeten Fragewörter liegt im Anhang bei. [24]

4.2 Baseline-Ansatz: Hidden-N-Gramm-Sprachmodell

Als Baseline dient ein rein textbasierter Ansatz: Es wird ein Hidden-N-Gramm-Sprachmodell aufgebaut, das als verstecktes Vokabular die Satzzeichen beinhaltet. Als Kontext n wird vier gewählt. Eine solche Kontextlänge hat den Vorteil, groß genug zu sein, um durch mehr Hintergrundwissen Fehlerraten reduzieren zu können aber nicht zu groß zu sein und somit noch relativ robust trainiert werden zu können. Im normalen Vokabular finden sich alle Wörter aus dem Vokabular des Spracherkenners wieder.
Das Sprachmodell wird somit auf reinen Texten mit Satzzeichen, ohne Prosodie oder Part-of-speech trainiert.

4.3 Entscheidungsbaum

Zum Vergleich mit dem Sprachmodell wird ein Entscheidungsbaum trainiert. Bezüglich der Frage, welche Werte in diesen aufgenommen werden sollten, wurden die folgenden Entscheidungen getroffen.

4.3.1 Entwurfsentscheidungen

Aus den Merkmalen, auf deren Basis der Entscheidungsbaum arbeitet, ergeben sich auch variable Parameter, die die Güte des Modells beeinflussen. In den folgenden Unterkapiteln werden die Merkmale beschrieben und ihre Wahl begründet.

4.3.1.1 Kontextgröße

Ein Parameter des Modells bestimmt die Größe des Kontextes und somit wie weit in die Vergangenheit und wie weit in die Zukunft bei der Entscheidung des Modells Merkmale betrachtet werden. Die Größe des Kontextes beeinflusst auch die Größe des resultierenden Baums, da jedes Merkmal, das in den Baum aufgenommen wird, in einem Baumknoten resultieren kann. In dieser Arbeit wird ein Kontext von 3 in die Vergangenheit und 1 in die Zukunft verwendet. Diese Wahl wird im Evaluationskapitel (6.2, S. 34) anhand von Tests mit Kontexten von bis zu sechs Wörtern in die Vergangenheit und von bis zu sechs Wörtern in die Zukunft begründet.

4.3.1.2 Anzahl der Wörter

Das gesamte Vokabular des Spracherkenners in den Entscheidungsbaum aufzunehmen, erweist sich als ineffizient sowohl bezüglich Speicherplatz als auch bezüglich Bearbeitungszeit. Dies deckt sich nicht mehr mit der Anforderung der Sprach- und Zeichenerkennung in Echtzeit. Dieser Ansatz wird in dieser Arbeit daher nicht weiter verfolgt.
Was die Aufnahme von Wörtern anbetrifft, verbleiben zwei Ansätze: Zum Einen können die Wörter komplett weggelassen werden, zum anderen können ausgewählte Schlüsselwörter aufgenommen werden und alle anderen vorkommenden Wörter auf ein Symbol „x“ abgebildet werden.
Als Schlüsselwörter dienen die Wörter, die in den Trainingsdaten am häufigsten vor oder nach einem Satzzeichen auftreten. Diese Wahl erweist sich deshalb als sinnvoll, weil diese Wörter auch im Test oftmals vor oder nach Satzzeichen erscheinen werden. Sie können daher ein möglichst geradliniges Erkennen von Satzzeichen ermöglichen. Die Zählung der auftretenden Worte kann sowohl absolut als auch relativ geschehen. Absolut bedeutet, dass lediglich gezählt wird, wie oft das Wort im Trainingstext vor oder nach einem Satzzeichen auftritt. Bei der relativen Zählweise wird diese Zahl noch durch die Gesamtzahl aller Auftritte des Wortes in den Trainingstexten dividiert. Auf diese Art führt eine Konjunktion, die fast immer nach einem Satzzeichen erscheint, zu einer höheren Zahl als ein Wort wie „you“, das zwar sehr oft im Kontext von Satzzeichen auftreten kann aber auch sehr oft ohne Satzzeichen.
Weitere Schlüsselwörter können den Quaero-Richtlinien für Zeichensetzung entnommen worden. Quaero ist ein französisches Projekt zur Verbesserung von Suchmaschinen inklusive der Entwicklung von Möglichkeiten zur Indizierung von Audio- und Videodateien. Um das Bewerten der verwendeten Spracherkennungssysteme einheitlich zu gestalten und damit vergleichen zu können, wurden Richtlinien entwickelt, auf welche Art die Zeichen in den Referenzen zu setzen sind. Aus diesen lassen sich

Schlüsselwörter extrahieren, die im Anhang aufgelistet sind.
In dieser Arbeit werden die 125 absolut häufigsten Schlüsselwörter in den Trainingstexten ohne weitere Quaero-Schlüsselwörter verwendet. Diese Wahl wird im Evaluationskapitel (6.2, S. 34) anhand von Tests mit verschiedenen Schlüsselwortanzahlen begründet.

4.3.1.3 Prosodische Merkmale

Der Entscheidungsbaum wird mit zwei prosodischen Merkmalen trainiert. Die verwendeten prosodischen Merkmale sind Pausenlänge hinter dem zu klassifizierenden Wort und eine Wortdauer-Wortlänge-Relation, die anzeigt, ob ein Sprecher ein Wort besonders langsam gesprochen hat. Dies kommt zwar eventuell bei schwer auszusprechenden Worten vor, oft aber auch vor Satzbegrenzungen.
Um auftretende Sprecherwechsel durch ein Satzzeichen unterstützen zu können, wird zudem aufgenommen, ob dem aktuell zu klassifizierenden Wort ein Sprecherwechsel folgt.

4.3.1.4 Wortarten (POS)

Der Entscheidungsbaum wird mit Part-of-speech-Tags trainiert. Sie werden genutzt, um Wörter zu generalisieren. Dadurch wird der Entscheidungsbaum klein gehalten, was auch die Berechnungen beschleunigt. Dies ist sehr wichtig im Bereich der Spracherkennung in Echtzeit.

4.3.1.5 Zugrundeliegende Trainingsdaten

Als Basis für die aufzunehmenden Wörter beziehungsweise Part-of-speech-Tags können die Wörter aus den Referenztexten oder die Wörter aus den Hypothesen des Spracherkenners dienen. Aus Gründen der Fehlerraten wird ein Baum auf Basis der Referenztexte verwendet. Ein Vergleich beider Varianten, der dies belegt, wird im Evaluationskapitel vorgestellt. (6.2, S. 34)

4.3.2 Resultierender Entscheidungsbaum

Die Eingaben des Entscheidungsbaums sehen wie folgt aus aus:

word(i-3) | pos(i-3) | word(i-2) | pos(i-2) | word(i-1) | tag(i-1) |
pos(i-1) | word(i) | pos(i) | relation | pauseLength | spkChange |
word(i+1) | pos(i+1)

Die Hypothese des Spracherkenners der Form

Wort	glued	to	the	TV	$(<BREATH>)	every
Dauer	0,25 s	0,08 s	0,11 s	0,48 s	0,22 s	0,29 s

führt beispielsweise für das Wort TV zu folgender Eingabe an den Entscheidungsbaum:

word(i-3)	pos(i-3)	word(i-2)	pos(i-2)	word(i-1)	tag(i-1)	pos(i-1)
x	V	to	NOTAG	the	NONE	DT
word(i)	pos(i)	relation	pauseLength	spkChange	word(i+1)	pos(i+1)
x	NN	0.240000	0.220000	FALSE	x	DT

Die verwendeten POS-Tags werden genauer im Kapitel 5 vorgestellt. Das Wort $(<BREATH>) ist ein Tag, der eine Atempause darstellt.

Am Entscheidungsbaum, der in Grafik 4.1 dargestellt ist, erkennt man, dass die

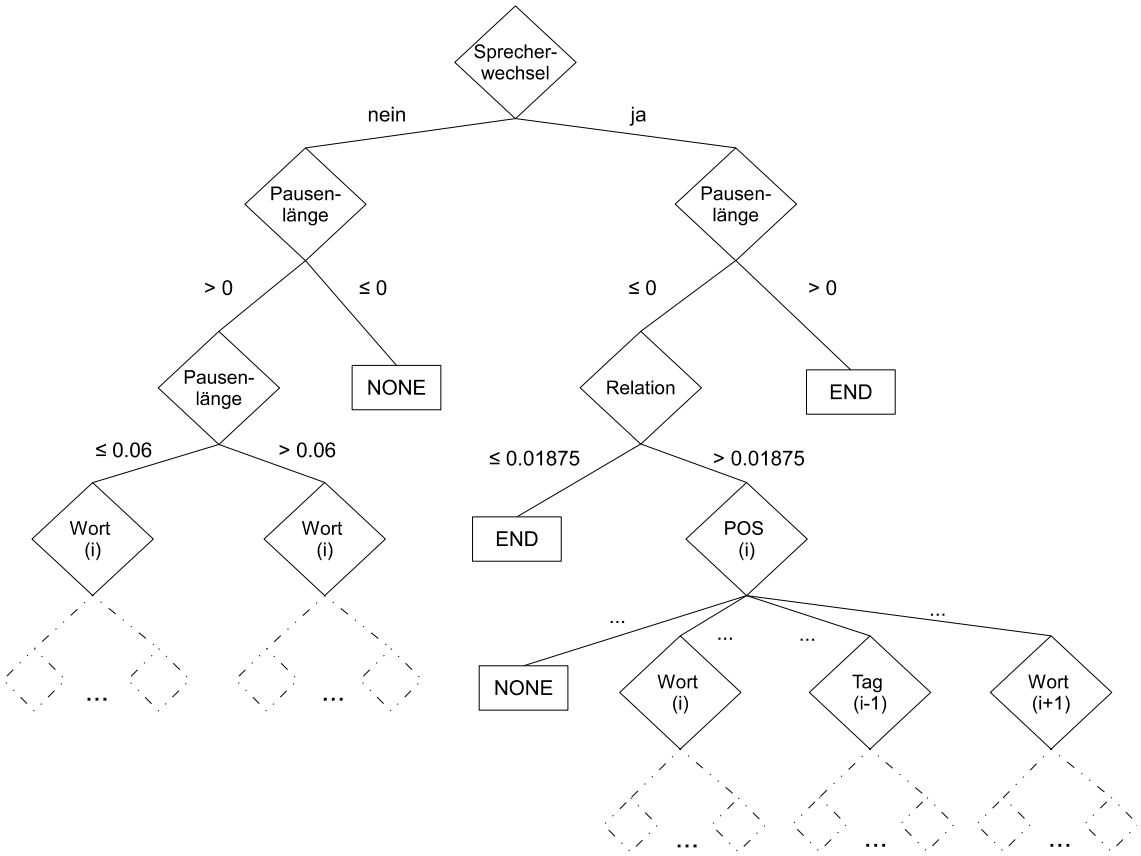

Abbildung 4.1: Ausschnitt des Entscheidungsbaums

hauptsächlichen Entscheidungen auf Basis der prosodischen Merkmale gefällt werden: Diese stehen ganz oben im Baum. Der Baum ist nur unvollständig dargestellt, da eine vollständige Darstellung den Platz sprengen würde.

4.4 Kombination der Modelle

Das Endsystem umfasst eine Kombination beider Modelle. Dazu werden die Modelle nicht mehr nur als Klassifizierer sondern auch als Wahrscheinlichkeitenschätzer verwendet. Die Wahrscheinlichkeiten werden kombiniert.

Bereits an den Ergebnissen der beiden Modelle allein wird deutlich, dass zu wenige Zeichen gesetzt werden. Aus diesem Grund wird im Endsystem die Wahrscheinlichkeit für die Klasse „NONE“ heruntergewichtet und die so freiwerdende Wahrscheinlichkeitsmasse gleichmäßig auf die beiden anderen Klassen „COMMA“ und „END“ verteilt. Für die Heruntergewichtung werden zwei verschiedene Arten getestet:

Zum Einen kann die Heruntergewichtung durch einen konstanten Faktor α erfolgen, zum Anderen durch eine lineare Gleichung in Abhängigkeit der Anzahl Wörter (k) seit dem letzten Satzzeichen:

$$\alpha(k) = 1 - (a \cdot k + b) \tag{4.1}$$

Hintergrund dieser Heruntergewichtung stellt die Beobachtung dar, dass Sätze und Nebensätze begrenzte Längen besitzen. Je mehr Worte in einem Satz vorhanden sind, desto mehr steigt daher die Wahrscheinlichkeit, bald ein Satzzeichen zu sehen. Die Formel zur Neuberechnung der Wahrscheinlichkeiten lässt sich wie folgt ausdrücken. Sie ergibt sich aus der Forderung, dass die Summe der Wahrscheinlichkeiten 1 ergeben soll. N stehe dabei für die Klasse „NONE“, C für die Klasse „COMMA“ und E für die Klasse „END“.

$$P(N)_{neu} = P(N)_{alt} \cdot \alpha_{Modell} \tag{4.2}$$

$$P(C)_{neu} = P(C)_{alt} + (1 - \alpha_{Modell}) \cdot P(N)_{alt} \cdot \frac{P(C)_{alt}}{P(C)_{alt} + P(E)_{alt}} \tag{4.3}$$

$$P(E)_{neu} = P(E)_{alt} + (1 - \alpha_{Modell}) \cdot P(N)_{alt} \cdot \frac{P(E)_{alt}}{P(C)_{alt} + P(E)_{alt}} \tag{4.4}$$

Aus diesen Formeln ergeben sich als Parameter:
$\alpha_{Sprachmodell}$, $\alpha_{Entscheidungsbaum}$, $a_{Sprachmodell}$, $b_{Sprachmodell}$ sowie $a_{Entscheidungsbaum}$ und $b_{Entscheidungsbaum}$.
Im Anschluss an die Heruntergewichtung werden Entscheidungsbaum und Sprachmodell durch lineare Interpolation kombiniert. Diese erfolgt nach der Formel:

$$P_{Kombination}(i) = \lambda \cdot P_{Entscheidungsbaum}(i) + (1 - \lambda) \cdot P_{Sprachmodell}(i) \text{ (vgl. [6])} \tag{4.5}$$

P gibt dabei die Wahrscheinlichkeit für Zeichen i an. Der Index gibt an, aus welchem Modell die Wahrscheinlichkeit stammt.
Die Kombination der Modelle birgt damit einen weiteren variablen Parameter des Systems: das Interpolationsgewicht λ.
In dieser Arbeit wird für das Sprachmodell eine konstante Heruntergewichtung und für den Entscheidungsbaum eine Heruntergewichtung mit Beachtung der Anzahl der Wörter seit dem letzten Satzende verwendet. Es ergeben sich folgende Formeln, deren Parameter im Evaluationsteil anhand von Tests verschiedener Gewichte begründet werden (6.2, S. 41):

- Heruntergewichtungsfaktor für das Sprachmodell:
  $\alpha_{Sprachmodell} = 0,8$
- Heruntergewichtungsfaktor für den Entscheidungsbaum:
  $\alpha_{Entscheidungsbaum}(k) = 0,6 \cdot k + 0,4$ (vgl. 4.1)
- Interpolation:
  $P(i) = 0,3 \cdot P_{Entscheidungsbaum}(i) + 0,7 \cdot P_{Sprachmodell}(i)$ (vgl. 4.5)

4.5 Zusammenfassung

Die Arbeit besteht damit aus folgenden Ansätzen, deren Implementierung und Ergebnisse in den nächsten Kapiteln beschrieben werden:

- Baseline-Ansatz: Hidden-4-Gramm-Sprachmodell: Modellierung von Worten
- Entscheidungsbaum: Modellierung von Schlüsselwörtern, POS und Prosodie
- Kombination von Hidden-4-Gramm-Sprachmodell und Entscheidungsbaum

5. Implementierung

In diesem Kapitel wird die konkrete Implementierung des Systems beschrieben. Die verwendeten Programme werden vorgestellt. Des Weiteren wird auf die Trainingsdaten der Modelle eingegangen.

5.1 Das Hidden-N-Gramm-Modell

Zum Bau des 4-Gramm-Sprachmodells wird das SRILM-Toolkit verwendet [25]. Als Trainingsmaterial dienen die in Tabelle 5.1 aufgelisteteten Texte. Die Tabelle gibt auch die Anzahlen der Satzzeichen innerhalb der Texte an. Für die Sprachmodellerstellung werden die vorkommenden Worte auf Kleinbuchstaben abgebildet. Durch diese Normalisierung kann das Modell robuster trainiert werden.
Das Smoothing erfolgt nach dem durch Chen und Goodman abgewandelten Kneser-Ney-Discounting. [26] Dies geschieht in Orientierung an die Sprachmodellerstellung für den verwendeten Spracherkenner. Zunächst werden für jeden Text einzeln Sprachmodelle erstellt und diese danach interpoliert. Die Interpolationsgewichte werden bei der N-Gramm-Erstellung für den Spracherkenner mit einem Skript berechnet, das die Perplexität des interpolierten Modells minimiert. Da das hier erstellte Sprachmodell aber nicht der Erkennung der Wörter dienen soll sondern der Erkennung der richtigen Satzzeichen, wurde dieses Skript so abgewandelt, dass es die Perplexität nicht anhand von Wortwahrscheinlichkeiten sondern anhand von Zeichenwahrscheinlichkeiten minimiert. Als Basis für die Berechnung der Zeichenwahrscheinlichkeiten wurde ein Teil der Quaero 2010 akustischen Trainingsdaten verwendet, der bei den Trainingstexten ausgespart wurde. Er umfasst 210.669 Wörter, 9.907 Punkte, 13.190 Kommata und 1.279 Fragezeichen.
Um das Sprachmodell effizient nutzbar zu gestalten, wird die geprunte Form in binärer Repräsentation verwendet. Diese Version enthält weniger Daten, da alle N-Gramme, deren Entfernung die Perplexität des Sprachmodells um weniger als eine bestimmte Schranke erhöht, weggelassen werden. Als Schranke wird in Orientierung an die Sprachmodellerstellung für den verwendeten Spracherkenner $1 \cdot 10^{-10}$ gewählt. Durch das Pruning reduziert sich somit der Speicherbedarf des Sprachmodells. Die Ladezeit des Sprachmodells wird außerdem durch die binäre Repräsentation verringert, da der Overhead des Parsens der Eingabe reduziert wird. Das Sprachmodell kann so effizient im Echtzeitbetrieb eingesetzt werden.

Tabelle 5.1: Trainingstexte für das Sprachmodell

Text	Wörter	Punkte	Kommata	Fragezeichen
Transkripte der akustischen Trainingsdaten von Quaero 2010	708.347	36.756 (4,61 %)	48.168 (6,04 %)	4.854 (0,61%)
Quaero 2010 Texttrainingsdaten	1.786.704.569	85.748.996 (4,46 %)	43.017.579 (2,24 %)	6.557.794 (0,34 %)
Central News Agency of Taiwan, 2009	33.116.390	1.547.342 (4,27 %)	1.549.826 (4,28 %)	2.706 (0,01 %)
Los Angeles Times/Washington Post, 2009	274.940.095	15.385.583 (5,04 %)	14.763.289 (4,83 %)	303.962 (0,10 %)
New York Times, 2009	1.495.325.898	88.878.669 (5,33 %)	82.315.576 (4,93 %)	1.755.973 (0,11 %)

5.2 Der Entscheidungsbaum

Als Entscheidungsbaumimplementierung wird C4.5 verwendet. Diese ist für diese Arbeit geeigneter als die Implementierung ID3, da sie die Möglichkeit bietet, Werte unbesetzt zu lassen und dennoch die wahrscheinlichsten Ausgaben berechnet. Der Grund hierfür wurde bereits in Kapitel 3.3, Seite 21 aufgeführt.
C4.5 erwartet als Eingaben für den Bau des Entscheidungsbaums zwei Dateien. In der einen Datei wird spezifiziert, welche Entscheidung zu treffen ist, das heißt, was auf den Blättern stehen wird, und welche Eingaben aus welchen Wertebereichen der Baum bekommen wird. In der anderen Datei stehen aufgelistet nach diesen Eingaben alle Trainingsdaten. Als Trainingsmaterial für den Entscheidungsbaum dienen die in Tabelle 5.2 aufgelisteten Texte inklusive der Transkripte ihrer Audiodateien. Die Tabelle gibt auch die Anzahlen der Satzzeichen innerhalb der Texte an. Für die Erstellung des Entscheidungsbaums werden alle vorkommenden Großbuchstaben auf Kleinbuchstaben abgebildet. Diese Normalisierung führt zu einem robusteren Training.
Für die Part-of-Speech-Tag-Bestimmung wird der auf einem Entscheidungsbaum basierende Part-of-speech-Tagger der Universität Stuttgart verwendet. [27] Im Englischen basiert dieser auf dem Penn-Treebank-Tagset. [28] Diese Tags werden jedoch nicht alle verwendet, um den Entscheidungsbaum erstens kleinzuhalten und ihn dadurch zweitens robuster trainieren zu können. Vielmehr werden beispielsweise alle Verben, die nicht in Gerundiumsform vorliegen, unabhängig ihrer Zeit oder Person auf denselben Tag abgebildet. Damit ergibt sich das in Tabelle 5.3 aufgelistete Tagset.

Tabelle 5.2: Trainingstexte für den Entscheidungsbaum

Text	Wörter	Punkte	Kommata	Fragezeichen
Zeichensetzungstrainingsdaten von Quaero 2011	276.304	13.762 (4,42 %)	19.323 (6,20 %)	2.108 (0,68 %)
Akustische Trainingsdaten von Quaero 2010	900.846	22.724 (2,38 %)	24.837 (2,61 %)	4.469 (0,47 %)

Tabelle 5.3: Verwendete POS-Tags

Tag	ursprüngliche Tags aus dem Penn-Treebank-Tagset	Bedeutung
NUM	CD, NUM	Zahlen
JJ	JJR, JJS, JJ	Adjektive in verschiedenen Steigerungsformen
NN	NNS, NN, PP	Nomen und Pronomen (mögliche Subjekte)
NP	NPS	Eigennamen im Singular oder Plural
DT	DT, PP$, PDT, POS	Artikel und besitzanzeigende Genitive (Wörter, die vor einem Subjekt kommen können)
FIL	RBR, RBS, RB, UH	Adverbien und Ausrufe (mögliche Füllwörter)
V	VBD, VBN, VBP, VBZ, VHD, VHN, VHP, VHZ, VVD, VVN, VVP, VVZ, VB, VH, VV	Verben in verschiedenen Zeitformen
VG	VVG, VHG, VBG	Verben in Gerundiumsformen
WP	WDT, WP$, WRB	Wörter, die mit wh beginnen, das heißt möglicherweise nach einem Satzzeichen kommen
MD	MD	Modalverben
SUB	IN, SUB/that	Präposition
CC	CC, IN	Konjunktionen
NOTAG	EX, FW, LS, RP, SENT, SYM, TO	restliche Wörter

5.3 Einbau in den Lecture Translator

Das entworfene System wird in den Lecture Translator eingebaut. [4] Es wird in die Sprachsegmentierungskomponente zwischen Spracherkennung und Beginn der Übersetzung geschaltet. Die Sprachsegmentierungskomponente „Language Segmenter“ sucht sich einen passenden Abschnitt des aufgenommenen Audiosignals und übergibt diesen an den Spracherkenner. Dieser liefert die wahrscheinlichste Wortsequenz zurück. Die Modelle des Zeichensetzungssystems erhalten diese erkannte Wortsequenz zusammen mit den Pausen- und Wortlängeninformationen als Eingabe. Die vermuteten Satzzeichen werden vor der Weitergabe an die Übersetzungskomponente und an die Ausgabekomponente in den Wortstrom als eigene Wörter eingefügt.

6. Evaluierung

Die Evaluierung des Systems erfolgt auf den Quaero-Development-Daten von 2010. Bezüglich der Zeichensetzung liegen zwei Versionen vor. Diese wurden von Muttersprachlern mit linguistischem Hintergrund erstellt. [29]
Tabelle 6.1 zeigt die Verteilung der Satzzeichen sowie die Gesamtzahl Worte des Textes.

Tabelle 6.1: Versionen der Development-Daten

Version	Worte	Punkte	Kommata	Fragezeichen
Version 1	39167	2088 (4,69 %)	3050 (6,85 %)	243 (0,55 %)
Version 2	39167	2074 (4,66 %)	3013 (6,77 %)	234 (0,53 %)

Version 1 enthält mehr Satzzeichen als Version 2. Das Inter-annotator agreement (IAA) bestimmt, wie genau zwei verschiedene Annotationen übereinstimmen. Im Fall der beiden Satzzeichenversionen zeigt ein Test, dass 4225 Zeichen übereinstimmen. Des Weiteren setzt die zweite Version 644 Zeichen, die Version 1 nicht setzt und Version 1 enthält 704 Zeichen, die Version 2 nicht enthält. An 452 Stellen steht zwar in beiden Versionen ein Satzzeichen, allerdings nicht das gleiche.
Bei der Bewertung der Satzzeichen werden sowohl beide Versionen getrennt und als auch kombiniert betrachtet. Die Kombination bewertet ein Zeichen als korrekt, wenn es von mindestens einer der Referenzen bestätigt wird. Dies erweist sich als sinnvoll, da sich die beiden Referenzen nicht an allen Stellen einig darüber sind, ob ein Zeichen gesetzt werden muss oder nicht. In diesem Kapitel werden der Übersichtlichkeit halber nur die Ergebnisse der Kombination vorgestellt, die anderen Evaluationsergebnisse befinden sich im Anhang.
Im ersten Schritt der Evaluierung werden Punkt und Fragezeichen als einheitliches Satzendezeichen bewertet, da auch die Modelle damit trainiert wurden. Das dadurch gefundene beste System wird im nächsten Schritt noch mit der Unterscheidung zwischen Punkt und Fragezeichen bewertet.
In diesem Kapitel werden die Testergebnisse der einzelnen Modelle vorgestellt: Begonnen wird mit dem Baseline-Ansatz Sprachmodell, gefolgt von den Entscheidungsbäumen. Der Test der Kombination beider Modelle schließt die Begründung zur

Wahl der Parameter ab. Ihm folgt eine Analyse sowie der Fragezeichentest mit Analyse.
Zur Evaluierung wird das LNE-Scoring-Tool verwendet. Es erstellt unter anderem eine Datei, die die gefundene Gegenüberstellung von Referenztext und Hypothese zeigt. Diese wird auf Basis der minimalen Editierdistanz zwischen Referenztext und Hypothesentext berechnet. Sie zeigt für jedes Wort der Hypothese auf, welchem Wort der Referenz es am ehesten zugeordnet werden kann. Anhand von dieser Datei wird für jedes Satzzeichen überprüft, ob es korrekt oder fälschlicherweise gesetzt wurde, ob es mit einem anderen Satzzeichen verwechselt oder ausgelassen wurde. Auf Basis dieser Werte werden die in Teil 2.4 auf Seite 13 vorgestellten Bewertungsmaßstäbe ausgerechnet. Da der Fokus dieser Arbeit auf der generellen Strukturierung des Textes durch Zeichen und nicht in erster Linie auf der Korrektheit der Zeichen selbst liegt, werden Entscheidungen für oder wider Parameter auf Basis der SU-Fehlerrate und nicht auf Basis der Slot-Error-Rate getroffen.

6.1 Evaluation des Sprachmodells

Die Ergebnisse des Tests des Hidden-N-Gramm-Sprachmodells sind in Tabelle 6.2 aufgelistet.
Man sieht, dass die Fehlerraten SER und SU-Errorrate nur knapp unter 100% liegen. Dies bedeutet, dass es zwar nicht schlechter ist, Zeichen mit dem Sprachmodell zu setzen als gar keine Zeichen zu setzen, dass es aber auch kaum besser ist.
An der Precision erkennt man, dass die gesetzten Zeichen ungefähr zur Hälfte vollkommen richtig sind, am Recall dagegen sieht man die große Schwäche des Systems: Es werden zu wenige Zeichen gesetzt. Nur 22,43% aller Satzbegrenzungen des Textes werden auch korrekt gesetzt.

Tabelle 6.2: Evaluation des Sprachmodells
Die Werte in den Zellen sind in Prozent angegeben.

Precision	57,00
SU-Precision	59,53
Recall	21,48
SU-Recall	22,43
SER	93,77
SU-Error	92,82

6.2 Evaluation des Entscheidungsbaums und seiner Parameter

Die variablen Parameter des Entscheidungsbaums, die im Entwurfskapitel 6.2.2 auf Seite 38 vorgestellt wurden, werden hier getestet und damit ihre Wahl begründet. Es handelt sich um drei Parameter: Kontextlänge, Schlüsselwortanzahl und ob man als Trainingsdaten Referenzwörter oder Hypothesenwörter nehmen sollte.
Da sich der Baum auf Basis der Hypothesendaten als etwas schlechter als der Baum auf Basis der Referenzdaten herausstellt (vergleiche 6.2.2, S. 38), werden die Ergebnisse der Kontext- und Schlüsselwort-Tests nur für letzteren dargestellt. Im Anschluss an die Tests erfolgt ein kurzer Vergleich mit dem Baum auf Hypothesendaten, um zu zeigen, dass dessen Werte schlechter sind. Die Testergebnisse dieses Baums liegen im Anhang bei.

6.2.1 Parameter: Kontextlänge

Der Entscheidungsbaum wird zunächst mit einer festen Schlüsselwortzahl (125 Wörter nach absoluter Zählweise ohne zusätzliche Wörter aus den Quaero-Richtlinien) auf verschiedene Kontexte getestet. Dabei ergeben sich die in den Tabellen 6.3, 6.4, 6.5 sowie in den Grafiken 6.1, 6.2, 6.3 dargestellten Ergebnisse.
Die Tabellen stellen die Precision-, Recall- und Fehlerraten bei den getesteten Kontexten dar. An den gefärbten Werten erkennt man die jeweils besten Bewertungen. Mit V wird die Kontextlänge in die Vergangenheit spezifiziert, mit Z die Kontextlänge in die Zukunft.

Tabelle 6.3: Precision-Ergebnisse des Baums bei verschiedenen Kontexten
Die Werte in den Zellen sind in Prozent angegeben.

		Z 1	Z 2	Z 3	Z 4	Z 5	Z 6
P	V 1	55,34	55,22	56,78	53,37	52,33	54,83
P_{SU}		74,15	**74,20**	74,13	69,68	68,01	71,39
P	V 2	53,93	55,03	54,17	53,54	54,93	52,56
P_{SU}		72,64	73,30	71,88	70,75	71,09	68,61
P	V 3	53,58	55,05	53,68	53,07	54,06	51,66
P_{SU}		74,12	73,76	72,21	68,42	70,66	67,93
P	V 4	52,73	54,22	53,61	54,97	**57,18**	49,87
P_{SU}		71,68	71,96	71,77	70,09	72,77	65,66
P	V 5	52,64	53,65	53,38	53,00	50,52	43,51
P_{SU}		71,55	71,83	71,52	68,44	65,04	57,38
P	V 6	52,47	53,57	53,28	50,82	44,67	36,16
P_{SU}		71,12	71,65	71,10	65,87	59,52	48,16

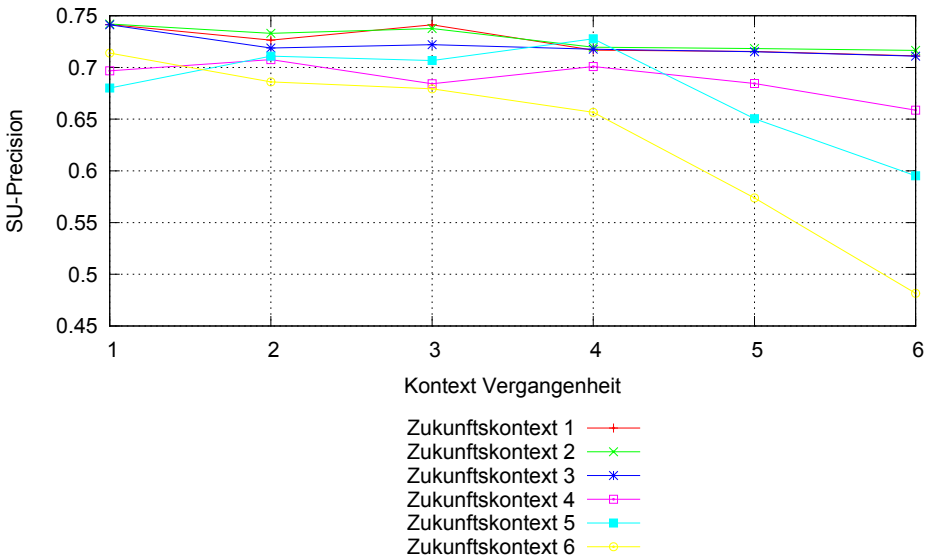

Abbildung 6.1: Precision-Ergebnisse des Kontexttests

Beim Betrachten der Tabelle 6.3 sowie der Abbildung 6.1 fällt schnell ins Auge, dass sich die Werte relativ ähneln. Nur bei sehr langen Kontexten gibt es einige schlechtere Ergebnisse. Dies zeigt sich in den nachfolgenden Tabellen und Abbildungen zu

den Kontexten ebenfalls. Es könnte damit zusammenhängen, dass sich längere Kontexte aufgrund der großen Anzahl der möglichen Belegungen der Werte schlechter trainieren lassen als kürzere.
Allgemein zeigt sich, dass die Precision der Satzgrenzenerkennung höher ist als die Precision der Satzzeichen. Dies ergibt sich aus der Tatsache, dass viele Satzgrenzen zwar richtig erkannt werden, dann aber das falsche Zeichen gesetzt wird.

Tabelle 6.4: Recall-Ergebnisse des Baums bei verschiedenen Kontexten
Die Werte in den Zellen sind in Prozent angegeben.

		Z 1	Z 2	Z 3	Z 4	Z 5	Z 6
R	V 1	10,62	11,60	11,66	10,85	9,04	9,23
R_{SU}		14,23	15,59	15,22	14,17	11,75	12,02
R	V 2	11,43	11,44	12,07	9,59	8,85	9,34
R_{SU}		15,39	15,24	16,02	12,68	11,45	12,19
R	V 3	11,82	11,73	11,87	10,22	9,15	9,19
R_{SU}		16,35	15,72	15,97	13,18	11,96	12,09
R	V 4	**12,21**	11,68	11,91	10,06	9,77	8,43
R_{SU}		16,59	15,51	15,95	12,83	12,43	11,09
R	V 5	12,20	11,78	11,99	10,08	9,21	7,32
R_{SU}		**16,59**	15,77	16,07	13,02	11,86	9,66
R	V 6	12,09	11,57	11,68	8,52	7,47	5,45
R_{SU}		16,39	15,48	15,59	11,04	9,95	7,26

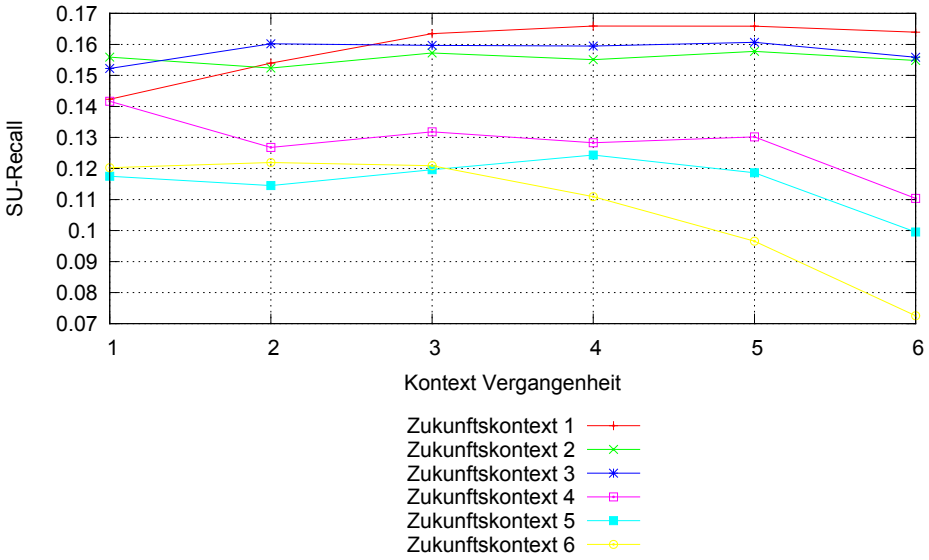

Abbildung 6.2: Recall-Ergebnisse des Kontexttests

Tabelle 6.4 und Abbildung 6.2 zeigen, dass die Recall-Werte sehr viel niedriger als die Precision-Werte sind. Auch sie werden bei höherem Kontext niedriger. Dies weist darauf hin, dass bei großem Kontext tendenziell erstens weniger Zeichen gesetzt werden (so kommen hohe Deletion-Werte und damit niedrige Recall-Ergebnisse zustande) und zweitens diese auch noch an falschen Stellen stehen (so ergeben sich niedrige Precision-Werte).
Analog zur Precision ist auch hier der Satzgrenzen-Recall höher als der Satzzeichen-Recall. Dies ergibt sich aus gleichem Grund wie bei den Precision-Werten. Dass der Abstand zwischen den beiden Werten nicht so hoch ist wie bei der Precision, kommt

durch den aufgrund der hohen Deletion-Zahl großen Nenner bei der Recall-Berechnung zustande.

Tabelle 6.5: Fehlerraten-Ergebnisse des Baums bei verschiedenen Kontexten
Die Werte in den Zellen sind in Prozent angegeben.

		Z 1	Z 2	Z 3	Z 4	Z 5	Z 6
SER	V 1	94,34	93,82	**93,65**	95,31	96,49	95,58
SU		90,73	89,83	90,09	92,00	93,78	92,79
SER	V 2	94,37	94,11	94,20	95,65	95,81	96,24
SU		90,40	90,31	90,25	92,56	93,21	93,39
SER	V 3	93,89	93,86	94,28	95,86	95,82	96,51
SU		**89,36**	89,87	90,18	92,90	93,01	93,62
SER	V 4	94,35	94,36	94,36	95,41	94,88	97,37
SU		89,96	90,53	90,33	92,64	92,22	94,71
SER	V 5	94,39	94,40	94,41	95,92	97,16	99,85
SU		90,01	90,41	90,33	92,98	94,51	97,52
SER	V 6	94,56	94,55	94,66	97,20	99,30	102,36
SU		90,27	90,64	90,75	94,68	96,82	100,55

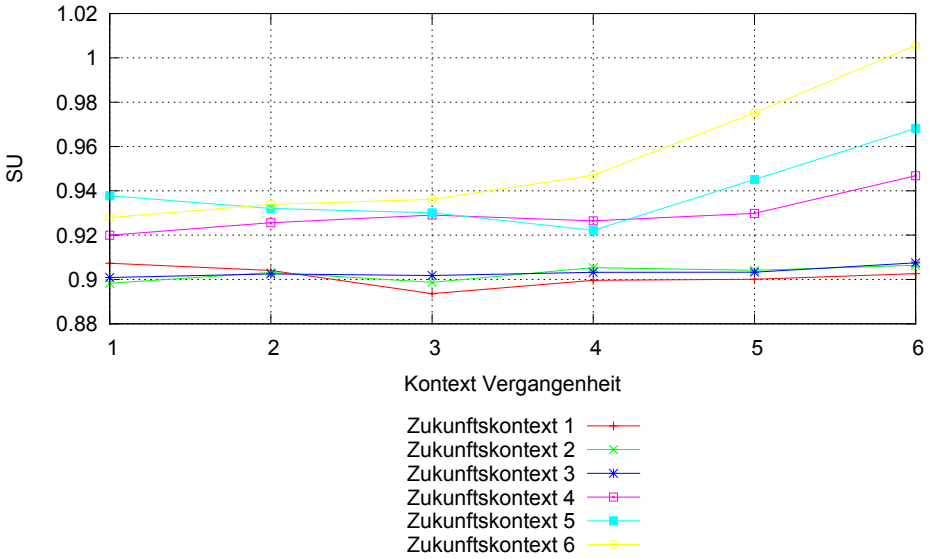

Abbildung 6.3: SU-Ergebnisse des Kontexttests

Analog zu den Precision- und Recall-Berechnungen zeigt sich auch in Tabelle 6.5 sowie in Abbildung 6.3, dass die Fehlerraten bei großen Kontexten deutlich höher sind als bei kleineren. Jedoch führt ein einseitig höherer Kontext in die Vergangenheit zu weniger hohen Fehlerraten als ein ebenso hoher Kontext in die Zukunft. Die ansonsten berechneten Fehlerraten erweisen sich als relativ ähnlich zueinander.
Die beste SU-Fehlerrate ergibt sich bei Kontext 3 in die Vergangenheit und Kontext 1 in die Zukunft. Der SER-Wert an dieser Stelle gleicht sehr dem von Kontext 3 in die Vergangenheit und Kontext 2 in die Zukunft. Jedoch ist die SU-Fehlerrate dort deutlich höher. Dies weist darauf hin, dass zwar weniger Satzzeichen verwechselt, dafür aber mehr andere Fehler begangen werden. Die Verwechslungen werden bei der SU-Fehlerrate schließlich nicht berücksichtigt.
Da zur Wahl des Parameters wie erwähnt die SU-Fehlerrate herangezogen wird,

ergibt sich als Ergebnis des Tests ein Kontext von 3 in die Vergangenheit und 1 in die Zukunft.
Der Baum auf Hypothesendaten erweist sich, wie aus den Tabellen und Grafiken im Anhang ersichtlich, mit einem Kontext von 1 in die Vergangenheit und 1 in die Zukunft als am besten.

6.2.2 Parameter: Schlüsselwortanzahl

Im nächsten Schritt werden der beste Referenzbaum und der beste Hypothesenbaum mit verschiedenen Schlüsselwörtern getestet. Wie im Entwurfskapitel beschrieben gibt es dabei drei Parameter: Die Anzahl der verwendeten Schlüsselwörter, ob diese absolut (a) oder relativ (r) gezählt werden und ob weitere Schlüsselwörter aus den Quaero-Richtlinien hinzugenommen werden (in den Tabellen mit + gekennzeichnet).
Die Ergebnisse des Referenzbaums sind in den Tabellen 6.6, 6.7, 6.8 sowie in den Grafiken 6.4, 6.5, 6.6 dargestellt, die Ergebnisse des Hypothesenbaums befinden sich im Anhang.
An den markierten Zahlen in den Tabellen erkennt man auch hier die je nach Bewertungsmaß besten Bäume.

Tabelle 6.6: Precision-Ergebnisse des Baums bei variabler Schlüsselwortanzahl
Die Werte in den Zellen sind in Prozent angegeben.

		0 Wörter	125 Wörter	250 Wörter	500 Wörter	1000 Wörter
P	a	49,80	53,58	51,87	51,83	52,96
P_{SU}		69,25	**74,12**	67,61	69,14	67,18
P	a +	48,71	**53,83**	52,54	51,75	50,34
P_{SU}		67,46	71,09	68,09	68,61	63,22
P	r	49,80	52,27	51,85	51,56	52,78
P_{SU}		69,25	70,96	68,61	69,97	69,16
P	r +	48,71	51,52	51,00	51,57	52,47
P_{SU}		67,46	68,73	68,33	69,37	68,76

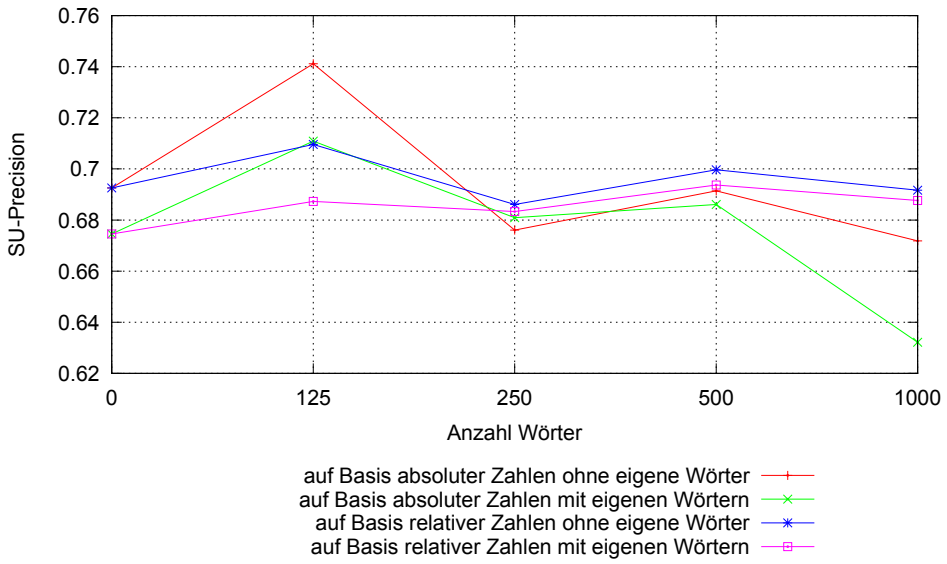

Abbildung 6.4: Precision-Ergebnisse des Schlüsselworttests

Tabelle 6.6 und Abbildung 6.4 zeigen, dass bei der absoluten Zählweise die Precision-Ergebnisse annähernd eine Kurve mit Höhepunkt bei 125 Wörtern ergeben. Der Grund für die besseren Werte bei Aufnahme eher weniger Wörter könnte darin liegen, dass viele Wörter im Training nicht oft genug gesehen werden, um für sie Wahrscheinlichkeiten robust und aussagekräftig schätzen zu können.
Bei relativer Zählweise schwanken die Werte eher unregelmäßig. Jedoch scheinen hier viele hilfreiche Schlüsselwörter erst mit höheren Wortanzahlen hinzuzukommen. Nur so könnte der Abfall der Precision-Raten, wie er bei der absoluten Zählweise zu beobachten ist, ausgeglichen werden.

Tabelle 6.7: Recall-Ergebnisse des Baums bei variabler Schlüsselwortanzahl
Die Werte in den Zellen sind in Prozent angegeben.

		0 Wörter	125 Wörter	250 Wörter	500 Wörter	1000 Wörter
R	a	10,71	11,82	10,85	10,16	7,97
R_{SU}		14,89	16,35	14,14	13,55	10,11
R	a +	10,01	10,53	10,91	10,30	9,28
R_{SU}		13,86	13,91	14,14	13,65	11,65
R	r	10,71	11,39	12,39	12,48	10,41
R_{SU}		14,89	15,47	16,40	16,94	13,64
R	r +	10,01	11,45	**12,87**	12,43	9,87
R_{SU}		13,86	15,27	**17,25**	16,73	12,93

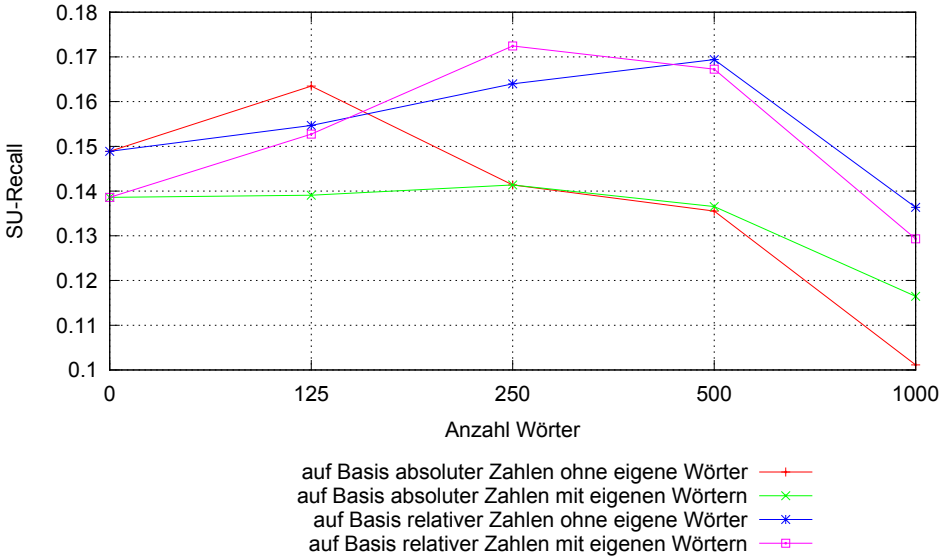

Abbildung 6.5: Recall-Ergebnisse des Schlüsselworttests

Während die absolute Zählweise bei der Precision bessere Ergebnisse liefert als die relative, zeigen Tabelle 6.7 und Abbildung 6.5, dass der Recall im Durchschnitt bei relativer Zählweise höher ist als bei absoluter. Er ergibt eine Kurve mit Höhepunkt bei 500 Wörtern.

Tabelle 6.8: Fehlerraten-Ergebnisse des Baums bei variabler Schlüsselwortanzahl
Die Werte in den Zellen sind in Prozent angegeben.

		0 Wörter	125 Wörter	250 Wörter	500 Wörter	1000 Wörter
SER	a	95,90	**93,89**	95,93	95,89	96,97
SU		91,72	**89,36**	92,64	92,50	94,83
SER	a +	96,68	95,12	95,72	95,95	97,50
SU		92,83	91,75	92,49	92,59	95,13
SER	r	95,90	94,94	95,11	94,79	95,67
SU		91,72	90,86	91,10	90,33	92,44
SER	r +	96,68	95,50	95,12	94,95	96,01
SU		92,83	91,68	90,75	90,66	92,94

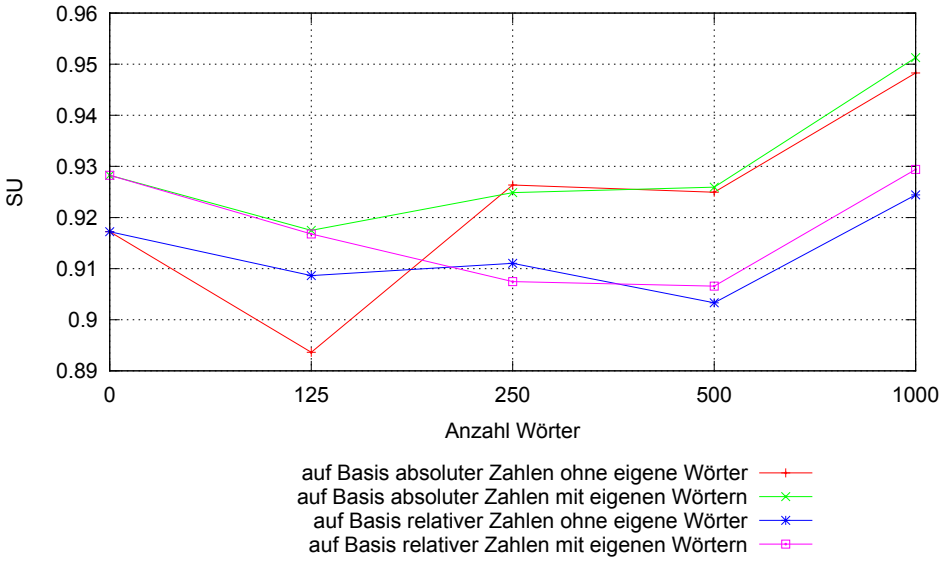

Abbildung 6.6: SU-Ergebnisse des Schlüsselworttests

In Tabelle 6.8 und in Abbildung 6.6 jedoch sieht man, dass die absolute Zählweise zu geringeren Fehlerraten führt als die relative. Die deutlich geringste Fehlerrate liegt bei 125 Schlüsselworten mit absoluter Zählweise vor.

6.2.3 Parameter: Trainingsdaten

Wie in den vorherigen Tests erwähnt, finden sich die genauen Ergebnisse des Baums auf Hypothesenwörtern im Anhang wieder. An dieser Stelle (Abbildung 6.7) zeigt eine Gegenüberstellung des besten Baums auf Referenzdaten mit dem besten Baum auf Hypothesendaten, dass der Baum auf Referenzdaten etwas besser abschneidet. Links in der Abbildung sind die zu maximierenden Bewertungsmaßstäbe dargestellt und rechts die zu minimierenden.

6.2.4 Ergebnis

Als bester Baum ergibt sich der Baum auf Referenzdaten mit Kontext 3 in die Vergangenheit und Kontext 1 in die Zukunft (vgl. 6.8) mit 125 Schlüsselwörtern absolut gezählt (vgl. 6.5). Dieser Baum wird für die folgenden Interpolationstests verwendet.

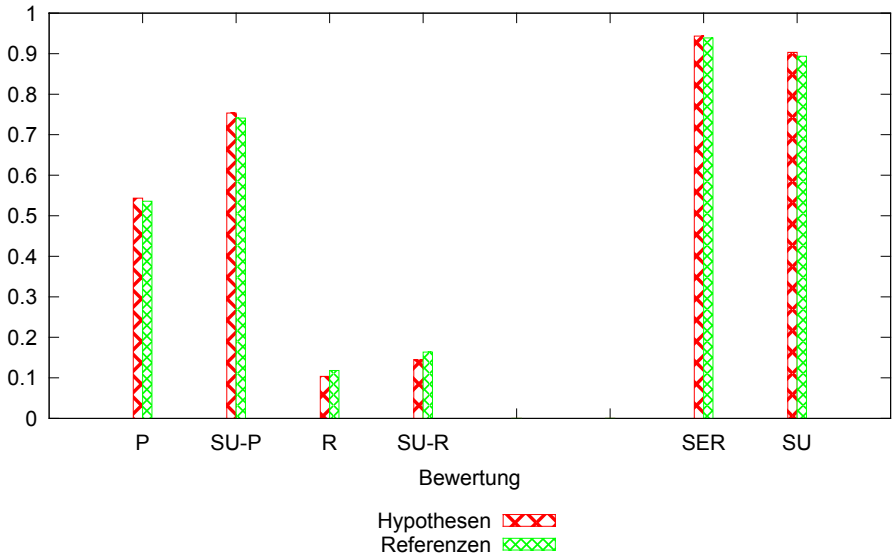

Abbildung 6.7: Baum auf Referenzdaten im Vergleich zum Baum auf Hypothesendaten

6.3 Kombination von Sprachmodell und Entscheidungsbaum

Mit dem gefundenen besten Baum wird die Interpolation mit dem Sprachmodell nach vorheriger Heruntergewichtung der NONE-Wahrscheinlichkeiten getestet.
Im ersten Unterkapitel erfolgt die NONE-Umgewichtung durch einen konstanten Faktor, im zweiten Unterkapitel durch eine lineare Gleichung. Die Interpolation schließlich wird bei gegebener bester Gewichtung im dritten Unterkapitel vorgestellt.
In den Tabellen der ersten beiden Unterkapitel sind jeweils die besten Werte für gegebene Gewichtung abgetragen, die sich aus nachfolgender Interpolation ergeben haben. Somit müssen die verwendeten Interpolationsgewichte zellenübergreifend nicht übereinstimmen. Die vollständigen Ergebnisse mit allen Heruntergewichtungsfaktoren und allen Interpolationsgewichten befinden sich im Anhang.

6.3.1 Ergebnisse mit NONE-Umgewichtung durch einen konstanten Faktor

Wie im Entwurf 4.4 beschrieben werden die Wahrscheinlichkeiten für die Klassifizierung der Wörter von Sprachmodell und Entscheidungsbaum umgewichtet und interpoliert.
Die Wahrscheinlichkeiten der Klasse NONE werden mit einem Faktor heruntergewichtet und die der anderen Klassen entsprechend angepasst, so dass sich als Summe der Wahrscheinlichkeiten wieder 1 ergibt. (Die genauen Formeln befinden sich in Kapitel 4.4, S. 26.) Für diesen Faktor werden alle Werte von 0,1 bis 1,0 in Zehntelschritten getestet. Die Heruntergewichtung mit 1,0 entspricht somit keiner Veränderung der Wahrscheinlichkeiten. Je niedriger der Faktor gewählt wird, desto stärker werden die Wahrscheinlichkeiten verändert.
Der Faktor für das Sprachmodell wird in den Tabellen mit L bezeichnet, der Faktor für den Entscheidungsbaum mit D.

Tabelle 6.9: Precision-Ergebnisse der Interpolation bezüglich verschiedener Herabgewichtungen der NONE-Wahrscheinlichkeiten
Die Werte in den Zellen sind in Prozent angegeben.

		L 0,1	L 0,2	L 0,3	L 0,4	L 0,5	L 0,6	L 0,7	L 0,8	L 0,9	L 1,0
P	D 0,1	11,54	11,54	11,54	11,54	16,54	32,38	35,65	37,01	38,02	40,66
P_{SU}		15,61	15,66	15,69	16,21	35,57	70,38	75,05	76,72	78,26	79,58
P	D 0,2	11,55	11,54	11,54	11,54	19,83	33,28	36,28	37,72	40,20	41,66
P_{SU}		15,68	15,59	15,67	16,59	43,12	71,23	75,11	77,06	78,34	79,64
P	D 0,3	11,57	11,55	11,54	11,54	24,07	34,11	37,05	38,77	41,10	41,96
P_{SU}		15,69	15,66	15,59	17,02	53,06	71,99	75,46	77,38	79,03	80,23
P	D 0,4	15,12	20,63	28,16	34,28	37,24	37,39	38,23	40,86	42,66	45,03
P_{SU}		20,07	27,99	40,84	51,65	62,98	72,87	75,92	78,48	80,14	81,79
P	D 0,5	37,75	38,28	38,97	40,58	41,61	43,09	44,32	45,30	46,28	47,35
P_{SU}		51,58	53,86	59,48	63,52	67,25	73,98	77,75	80,12	81,51	82,14
P	D 0,6	41,99	43,19	43,84	44,18	45,17	45,75	47,33	48,75	51,78	53,76
P_{SU}		59,42	60,68	63,18	66,86	69,74	75,19	79,45	81,21	82,34	83,37
P	D 0,7	45,33	45,77	45,96	46,30	47,81	49,31	50,85	54,46	56,15	59,36
P_{SU}		63,22	63,56	65,14	68,66	72,58	78,29	80,84	82,60	84,66	85,56
P	D 0,8	48,56	49,49	49,80	49,80	50,28	52,46	56,79	60,09	61,18	62,52
P_{SU}		67,17	68,16	68,61	70,61	74,60	79,97	82,73	84,45	85,30	86,48
P	D 0,9	50,33	50,76	51,51	51,82	52,92	55,23	60,14	61,55	63,09	64,00
P_{SU}		69,32	70,00	70,97	72,36	77,16	82,74	84,83	85,92	86,41	**87,29**
P	D 1,0	52,78	53,50	53,66	54,12	54,21	58,48	60,91	62,66	**64,40**	54,27
P_{SU}		73,24	74,01	74,26	74,46	78,80	83,40	85,42	86,62	87,19	74,78

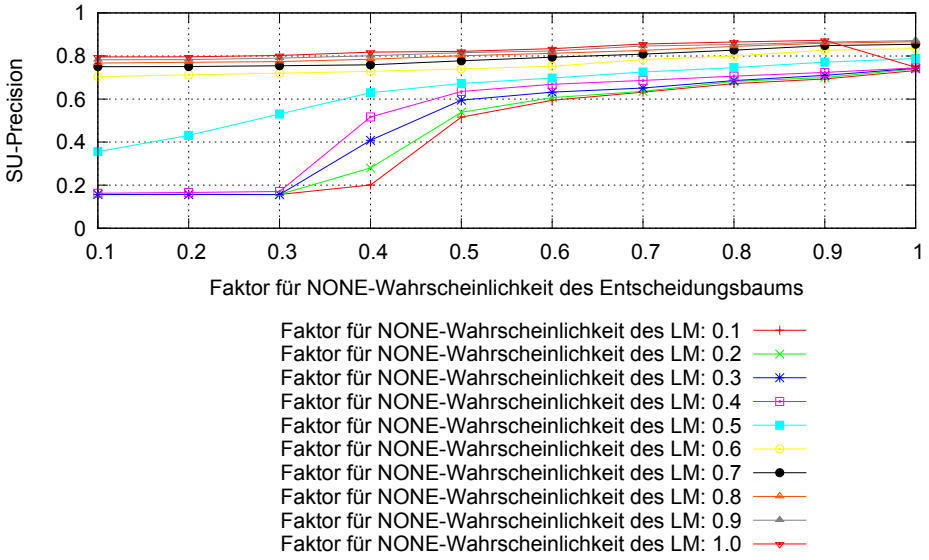

Abbildung 6.8: Precision-Ergebnisse der Umgewichtung durch einen konstanten Faktor

Tabelle 6.9 und Abbildung 6.8 deuten darauf hin, dass die Precision sinkt, je mehr die Wahrscheinlichkeiten verändert werden. Die höchste Precision ergibt sich bei keinerlei Veränderung der Werte, das heißt, bei einem Gewichtungsfaktor von 1,0. Dies lässt sich dadurch erklären, dass durch starke Heruntergewichtung der NONE-

Klassenwahrscheinlichkeit viel mehr Satzzeichen und diese an falsche Stellen gesetzt werden.

Tabelle 6.10: Recall-Ergebnisse der Interpolation bezüglich verschiedener Herabgewichtungen der NONE-Wahrscheinlichkeiten
Die Werte in den Zellen sind in Prozent angegeben.

		L 0,1	L 0,2	L 0,3	L 0,4	L 0,5	L 0,6	L 0,7	L 0,8	L 0,9	L 1,0
R	D 0,1	70,97	70,92	70,92	70,95	70,99	70,94	70,92	70,90	70,89	70,89
R_{SU}		95,85	95,90	**95,93**	95,77	95,66	95,52	94,60	94,56	94,54	94,42
R	D 0,2	70,99	70,94	70,92	70,92	70,97	70,95	70,92	70,90	70,89	70,89
R_{SU}		95,82	95,78	95,71	95,76	95,54	94,74	94,50	94,39	94,37	94,33
R	D 0,3	**71,10**	70,99	70,95	70,92	70,95	70,97	70,95	70,91	70,87	70,86
R_{SU}		95,67	95,62	95,55	95,58	94,46	94,39	94,29	94,28	94,26	94,24
R	D 0,4	65,78	57,77	52,90	46,24	33,81	31,13	29,43	28,50	27,78	27,07
R_{SU}		95,60	95,57	95,14	92,25	68,17	52,34	45,91	42,32	40,03	38,96
R	D 0,5	53,33	51,81	49,76	45,55	29,87	23,61	20,81	18,86	17,83	17,35
R_{SU}		95,15	95,11	95,09	91,46	65,82	49,80	41,03	35,41	31,60	28,26
R	D 0,6	51,01	50,15	49,53	44,63	28,98	22,57	18,41	16,14	14,76	14,58
R_{SU}		95,13	95,11	95,09	90,21	63,74	48,59	40,14	34,94	31,23	27,84
R	D 0,7	50,03	49,71	49,36	43,63	28,23	21,91	18,01	15,85	14,03	13,30
R_{SU}		95,11	95,07	95,06	88,96	61,94	47,41	39,23	34,19	30,66	27,08
R	D 0,8	49,69	49,52	49,09	42,65	27,37	21,40	17,69	15,53	13,51	12,63
R_{SU}		95,09	95,07	94,93	87,83	60,04	46,35	38,47	33,73	29,93	26,68
R	D 0,9	49,62	49,39	48,77	41,47	26,56	20,95	17,37	15,28	13,27	11,93
R_{SU}		95,07	95,07	94,76	86,40	58,27	45,23	37,77	33,18	29,39	26,09
R	D 1,0	49,31	49,34	48,41	40,20	25,90	20,59	17,05	14,98	12,99	11,93
R_{SU}		95,07	95,07	94,42	84,79	56,81	44,29	36,93	32,58	28,83	16,43

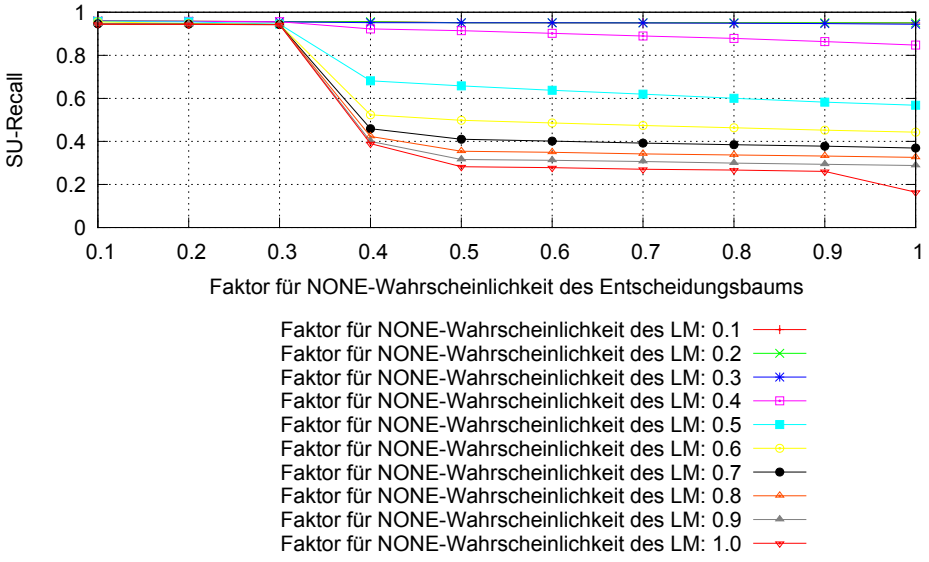

Abbildung 6.9: Recall-Ergebnisse der Umgewichtung durch einen konstanten Faktor

Tabelle 6.10 und Abbildung 6.9 zeigen, dass der Recall genau dort hoch ist, wo die Precision niedrige Ergebnisse geliefert hat: Je niedriger der Gewichtungsfaktor, das heißt je stärker die NONE-Klassenwahrscheinlichkeit geändert wird, ein desto höherer Recall-Wert ergibt sich. Dieses Ergebnis lässt sich gut erklären: Der Recall bewertet nicht, ob zu viele Zeichen gesetzt wurden, sondern nur wie viele der tatsächlichen Zeichen auch erkannt wurden. Werden aber von einem System, dessen Wahrscheinlichkeit für die Klasse NONE sehr gering ist, an fast allen Stellen Zeichen gesetzt, werden damit auch fast alle tatsächlichen Zeichen getroffen. Tabelle

Tabelle 6.11: Fehlerraten-Ergebnisse der Interpolation bezüglich verschiedener Herabgewichtungen der NONE-Wahrscheinlichkeiten
Die Werte in den Zellen sind in Prozent angegeben.

		L 0,1	L 0,2	L 0,3	L 0,4	L 0,5	L 0,6	L 0,7	L 0,8	L 0,9	L 1,0
SER	D 0,1	549,50	549,57	549,57	537,41	203,31	97,89	93,53	92,57	91,57	89,54
SU		520,96	520,71	518,96	491,74	162,26	68,03	66,96	**66,67**	69,15	67,38
SER	D 0,2	549,17	549,52	549,57	523,05	163,09	97,06	93,18	91,95	90,33	**89,53**
SU		517,99	519,80	519,45	477,07	123,44	67,69	67,87	68,50	68,36	70,12
SER	D 0,3	549,35	549,47	549,52	507,53	130,30	95,51	92,50	91,54	90,21	90,44
SU		518,02	518,35	520,36	461,35	91,90	67,23	69,06	70,41	71,04	72,38
SER	D 0,4	382,09	240,92	151,20	117,36	103,90	94,53	92,14	90,69	89,93	89,62
SU		360,53	220,73	132,32	95,63	74,46	67,85	70,21	71,90	73,34	75,02
SER	D 0,5	106,23	105,25	102,93	100,66	97,34	93,19	91,32	90,46	90,59	90,48
SU		98,15	95,16	84,82	77,67	71,38	68,35	71,67	74,35	76,60	78,77
SER	D 0,6	99,44	98,52	98,04	97,15	95,91	92,58	91,47	91,05	90,64	91,04
SU		92,48	89,96	81,60	76,32	69,71	68,82	72,40	74,88	76,86	79,04
SER	D 0,7	97,28	97,09	97,00	96,30	94,51	91,71	91,39	91,09	91,26	91,51
SU		91,58	85,80	78,61	74,94	68,56	69,30	72,94	75,40	77,17	79,53
SER	D 0,8	95,65	95,27	95,14	95,14	93,99	91,96	91,63	91,73	92,16	92,03
SU		90,50	83,52	79,77	74,45	68,35	69,63	73,17	75,44	77,70	79,66
SER	D 0,9	95,01	94,80	94,51	94,38	93,01	91,71	91,81	92,12	92,32	92,32
SU		90,19	85,57	77,84	72,98	69,05	70,07	73,30	75,69	78,04	80,13
SER	D 1,0	94,08	93,87	93,81	93,64	92,58	92,29	92,23	92,42	92,24	93,62
SU		89,42	84,85	80,45	73,28	68,63	70,45	73,70	76,01	78,40	89,11

6.11 sowie die Abbildung 6.10 bestätigen die Precision-Ergebnisse: Je geringer der Gewichtungsfaktor für die Klasse NONE ist, desto mehr falsche Zeichen werden gesetzt. So ergeben sich Fehlerraten von über 100 %. Der Graph 6.10 visualisiert die nicht allzu großen aber durchaus vorhandenen Schwankungen der restlichen Fehlerraten. In der Grafik zur Sentence-Unit-Errorrate (6.10) zeigt sich, dass die Werte für einen niedrigen Sprachmodell- und einen niedrigen Entscheidungsbaumfaktor sehr hoch sind (über 100 % Fehler). Für höhere Sprachmodellfaktoren sind die Fehlerraten unabhängig des Entscheidungsbaumfaktors einigermaßen moderat konstant, für niedrigere Sprachmodellfaktoren ergeben sich erst bei höheren Entscheidungsbaumfaktoren ebenso moderate Fehlerraten. Dies zeigt, dass mindestens einer der Heruntergewichtungsfaktoren hoch genug sein muss, um die Wahrscheinlichkeiten des Modells nicht allzusehr verändern. Allerdings zeigt sich auch, dass der Faktor nicht 1,0 sein darf, da dort die Fehlerraten nicht am niedrigsten sind.

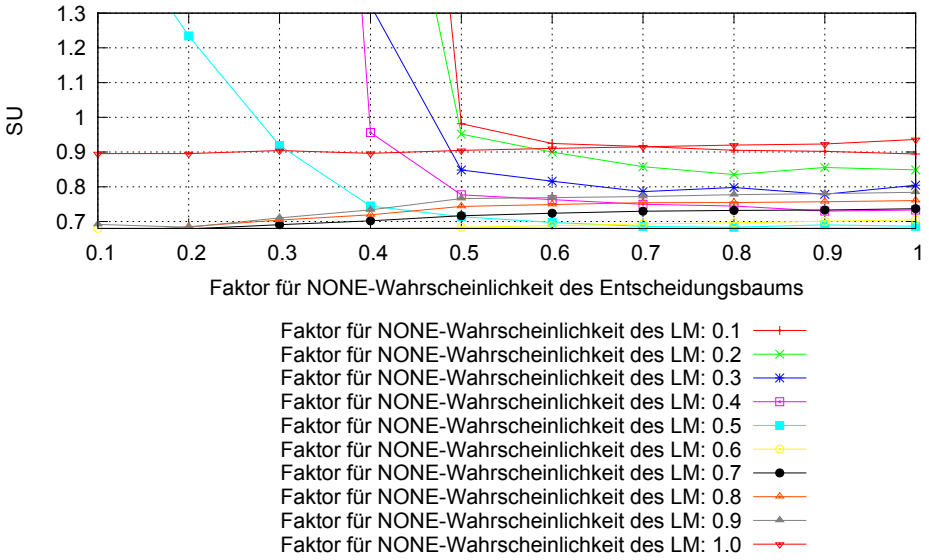

Abbildung 6.10: SU-Ergebnisse der Umgewichtung durch einen konstanten Faktor

Die besten Werte für die Heruntergewichtung ergeben sich bei einem Faktor für den Entscheidungsbaum von 0,1 und bei einem Faktor für das Sprachmodell von 0,7. Dies bedeutet, dass der Entscheidungsbaum so gut wie immer Zeichen setzt, während das Sprachmodell moderat umgewichtet wird.

6.3.2 Ergebnisse mit NONE-Umgewichtung durch eine lineare Gleichung

Wie im Entwurfskapitel (4.4, S. 26) beschrieben erfolgt die Umgewichtung im Endsystem nur für das N-Gramm-Sprachmodell durch einen konstanten Faktor. Die Wahrscheinlichkeiten des Entscheidungsbaums dagegen werden mit einem Wert gewichtet, der durch eine lineare Gleichung bestimmt wird, die von der Anzahl der bisher gesehenen Worte seit dem letzten Satzende abhängt:

$$\alpha_{Entscheidungsbaum} = 1 - (a \cdot k + b) \text{ (vgl. 4.1)}$$

In diesem Abschnitt wird gezeigt, dass dies für den Entscheidungsbaum bessere Ergebnisse liefert.

Tabelle 6.12 und Abbildung 6.11 zeigen, dass die Precision-Werte sinken, wenn die Werte a und b steigen. Dies deckt sich mit den Ergebnissen der Umgewichtung durch einen konstanten Faktor: Je höher die Werte a und/oder b sind, desto niedriger ist $\alpha_{Entscheidungsbaum}$. Damit erfolgt eine stärkere Umgewichtung. Sowohl im Falle der Umgewichtung durch einen konstanten Faktor als auch durch diese lineare Gleichung erweist sich die Precision als besser, je weniger umgewichtet wird.

Tabelle 6.12: Precision-Ergebnisse der Interpolation nach Herabgewichtungen der NONE-Wahrscheinlichkeiten durch lineare Gleichung 4.1

Die Werte in den Zellen sind in Prozent angegeben.

		b 0,0	b 0,2	b 0,4	b 0,6	b 0,8	b 1,0
P	a 0,0	**64,22**	62,08	53,76	43,18	41,66	37,57
P_{SU}		**88,24**	86,48	83,37	80,94	79,64	79,18
P	a 0,2	38,20	38,13	37,45	36,99	36,66	36,23
P_{SU}		79,24	79,19	78,86	78,59	78,65	78,58
P	a 0,4	38,01	37,62	37,26	36,90	36,36	36,23
P_{SU}		78,87	78,91	78,63	78,67	78,67	78,58
P	a 0,6	37,61	37,44	37,22	36,58	36,36	36,23
P_{SU}		78,89	78,77	78,66	78,69	78,67	78,58
P	a 0,8	37,52	37,43	36,94	36,58	36,36	36,23
P_{SU}		78,70	78,80	78,68	78,69	78,67	78,58
P	a 1,0	37,52	37,17	36,94	36,58	36,36	36,23
P_{SU}		78,74	78,81	78,68	78,69	78,67	78,58

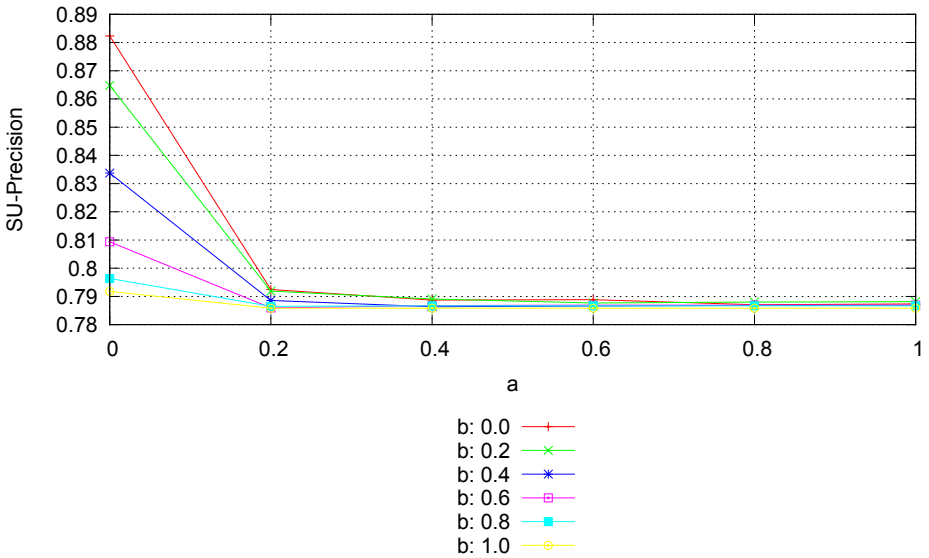

Abbildung 6.11: Precision-Ergebnisse der Umgewichtung durch eine lineare Gleichung

Auch Tabelle 6.13 und Abbildung 6.12 bestätigen die Ergebnisse der Heruntergewichtung durch einen konstanten Faktor: Je stärker die Umgewichtung, desto bessere Ergebnisse liefert der Test bezüglich des Recalls. Diese stärkere Umgewichtung wird durch ein niedriges $\alpha_{Entscheidungsbaum}$ erreicht, das sich bei hohen Werten von a und/oder b ergibt.

Tabelle 6.13: Recall-Ergebnisse der Interpolation nach Herabgewichtungen der NONE-Wahrscheinlichkeiten durch lineare Gleichung 4.1

Die Werte in den Zellen sind in Prozent angegeben.

		b 0,0	b 0,2	b 0,4	b 0,6	b 0,8	b 1,0
R	a 0,0	49,49	49,69	50,79	70,63	71,07	**71,30**
R_{SU}		95,07	95,11	95,13	95,57	95,93	**95,94**
R	a 0,2	49,49	52,92	65,54	70,67	71,07	**71,30**
R_{SU}		95,07	95,11	95,13	95,57	95,93	**95,94**
R	a 0,4	53,46	65,22	66,21	70,65	71,12	**71,30**
R_{SU}		95,07	95,11	95,13	95,59	95,93	**95,94**
R	a 0,6	64,90	65,87	66,44	69,66	71,12	**71,30**
R_{SU}		95,07	95,11	95,13	95,59	95,93	**95,94**
R	a 0,8	65,64	66,18	50,79	69,66	71,12	**71,30**
R_{SU}		95,07	95,11	95,13	95,59	95,93	**95,94**
R	a 1,0	66,06	49,69	50,79	69,66	71,12	**71,30**
R_{SU}		95,07	95,11	95,13	95,59	95,93	**95,94**

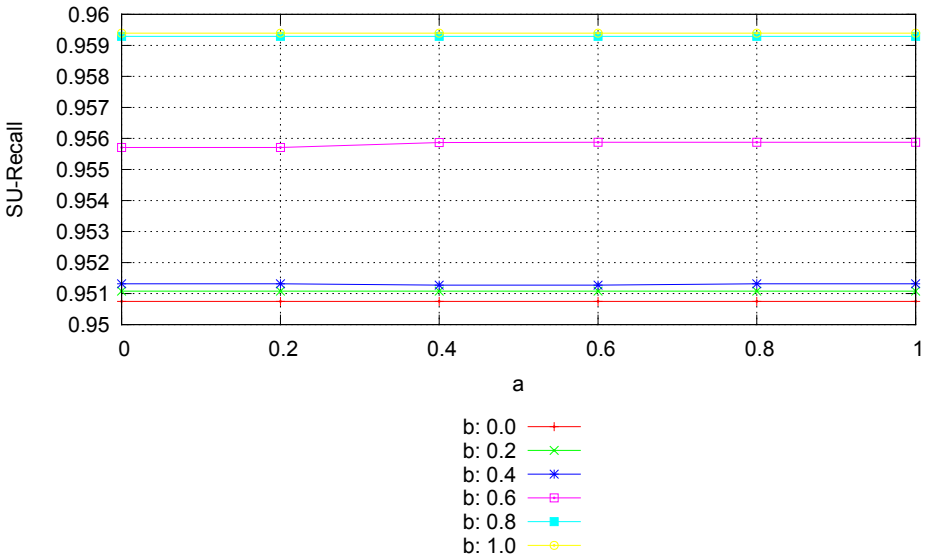

Abbildung 6.12: Recall-Ergebnisse der Umgewichtung durch eine lineare Gleichung

Ebenso analog zur konstanten Heruntergewichtung hat der SU-Bewertungsmaßstab seine besten Werte bei einer starken Umgewichtung des Entscheidungsbaums. Die besten Ergebnisse kommen bei $a > 0,4$ zu Stande. Je höher a gewählt wird, desto stärker fällt die Anzahl der bisher gesehenen Worte seit dem letzten Satzende ins Gewicht. Die Umgewichtung wird bei hohem a mit wachsender Wortanzahl schnell größer, da die Wortanzahl mit a multipliziert wird.

Tabelle 6.14: Fehlerraten-Ergebnisse der Interpolation nach Herabgewichtungen der NONE-Wahrscheinlichkeiten durch lineare Gleichung 4.1

Die Werte in den Zellen sind in Prozent angegeben.

		b 0,0	b 0,2	b 0,4	b 0,6	b 0,8	b 1,0
SER	a 0,0	92,42	91,79	91,05	90,69	**89,53**	92,90
SU		70,45	69,63	68,82	68,20	68,51	68,04
SER	a 0,2	91,79	91,76	92,17	92,36	92,52	93,04
SU		67,38	67,29	66,36	66,16	66,01	66,13
SER	a 0,4	91,94	92,03	92,16	92,31	92,73	93,04
SU		66,60	66,39	66,13	66,08	66,07	66,13
SER	a 0,6	91,93	92,06	92,12	92,54	92,73	93,04
SU		66,47	66,29	**65,95**	66,14	66,07	66,13
SER	a 0,8	91,99	91,98	92,31	92,54	92,73	93,04
SU		66,30	66,10	66,05	66,14	66,07	66,13
SER	a 1,0	91,91	92,17	92,31	92,54	92,73	93,04
SU		66,12	66,27	66,05	66,14	66,07	66,13

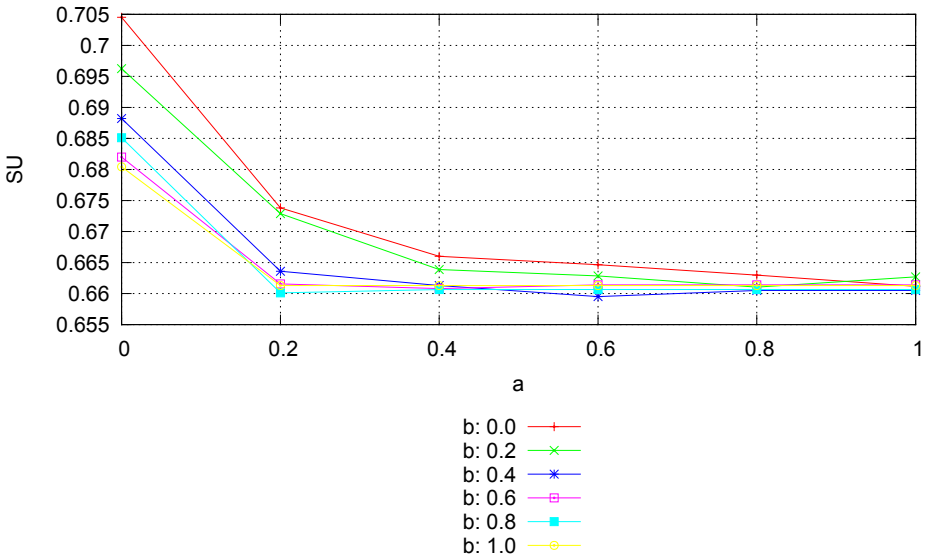

Abbildung 6.13: SU-Ergebnisse der Umgewichtung durch eine lineare Gleichung

Umgewichtung des Sprachmodells

Abbildung 6.14 zeigt deutlich, dass das Sprachmodell die besten Ergebnisse bei $c = 0$ liefert. Bei einer Berechnung des Heruntergewichtungsfaktors mit

$$\alpha_{Sprachmodell} = 1 - (c \cdot k + d)$$

ist dies gleichbedeutend mit $\alpha_{Sprachmodell} = 1 - d$. Der Faktor bleibt damit unabhängig von der Anzahl der bisher gesehenen Worte seit dem letzten Satzende konstant.

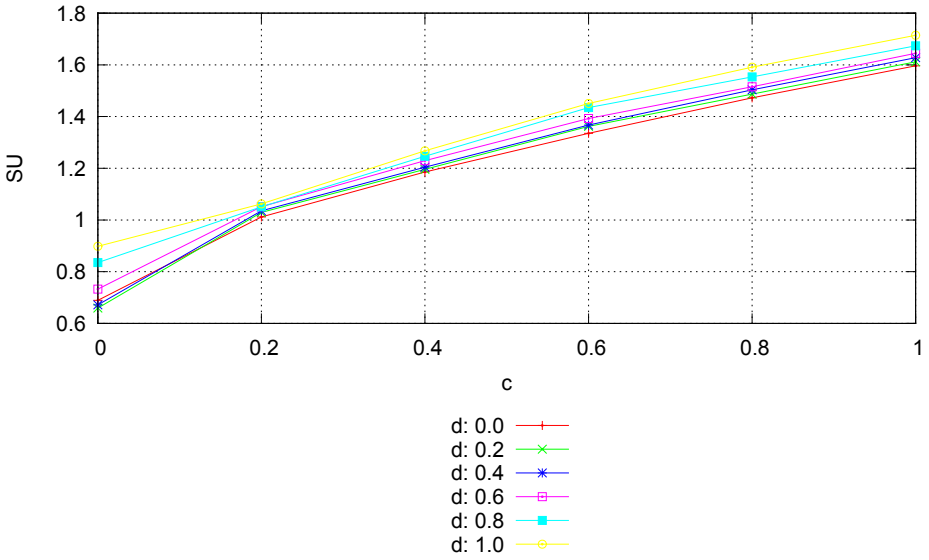

Abbildung 6.14: SU-Ergebnisse der Umgewichtung des Sprachmodells durch eine lineare Gleichung

6.3.3 Interpolation nach Umgewichtung der Wahrscheinlichkeiten

Wie im Entwurf beschrieben und in den vorhergehenden Unterkapiteln gezeigt, finden sich die besten Ergebnisse, wenn das Sprachmodell mit dem konstanten Faktor $0,8$ und der Entscheidungsbaum mit dem Ergebnis der linearen Gleichung $1-(0,6 \cdot k+0,4)$ umgewichtet wird. Die Tabellen 6.15, 6.16 und 6.17 die Abbildungen 6.15, 6.16 und 6.17 zeigen dort die Ergebnisse für verschiedene Interpolationsgewichte.

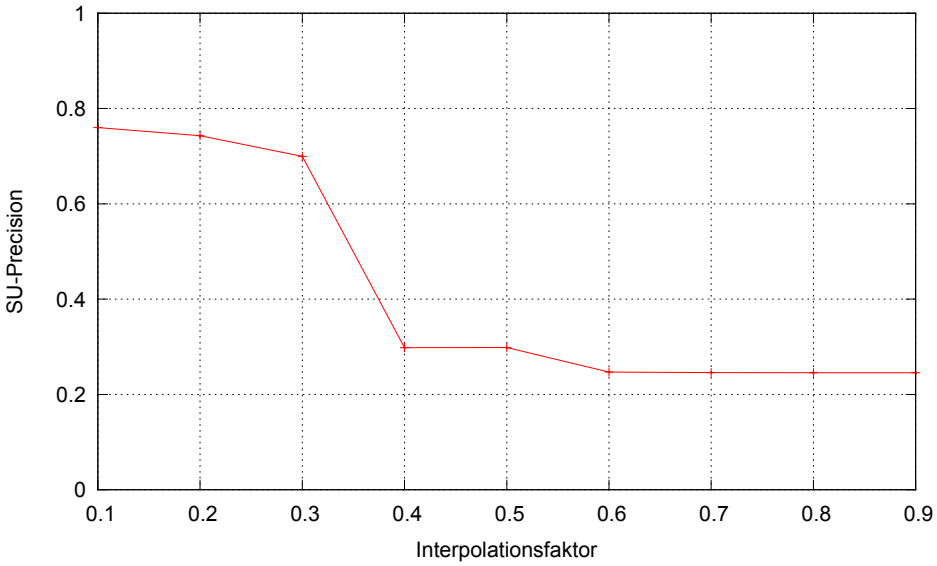

Abbildung 6.15: Precision-Ergebnisse der Interpolation

Tabelle 6.15: Precision-Ergebnisse der Interpolation
In den Spalten steht der Faktor der Interpolation. Die Werte in den Zellen sind in Prozent angegeben.

	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
P	35,38	**35,47**	33,53	16,56	17,49	18,31	19,26	19,84	19,98
P_{SU}	**76,01**	74,31	69,96	29,81	29,84	24,68	24,58	24,54	24,54

Dass die höchste Precision bei 0,1 liegt, passt zu der Beobachtung, dass die Precision höher ist, je weniger die Wahrscheinlichkeiten verändert werden: Da die Ausgaben des Sprachmodells weniger umgewichtet werden als die des Entscheidungsbaums, muss für eine gute Precision das Sprachmodell zu höchstmöglichem Anteil einfließen. Der große Unterschied in der zweiten Hälfte der Tabelle gegenüber der ersten kommt dadurch zu Stande, dass ab einem Interpolationswert von 0,5 der stark umgewichtete Entscheidungsbaum zu einem höheren Anteil einfließt als das moderat umgewichtete Sprachmodell.

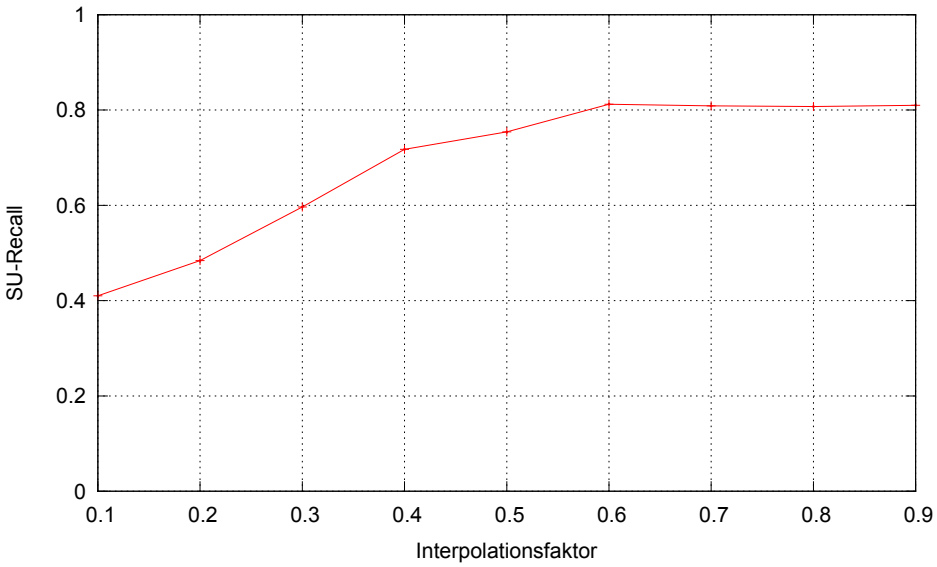

Abbildung 6.16: Recall-Ergebnisse der Interpolation

Tabelle 6.16: Recall-Ergebnisse der Interpolation
In den Spalten steht der Faktor der Interpolation. Die Werte in den Zellen sind in Prozent angegeben.

	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
R	19,09	23,10	28,59	39,85	44,20	60,24	63,37	65,28	**65,92**
R_{SU}	41,00	48,40	59,66	71,75	75,42	**81,22**	80,88	80,75	80,98

Analog zu den bisherigen Tabellen und Abbildungen ergeben sich auch in Tabelle 6.16 für den Recall genau gegensätzliche Ergebnisse wie für die Precision.

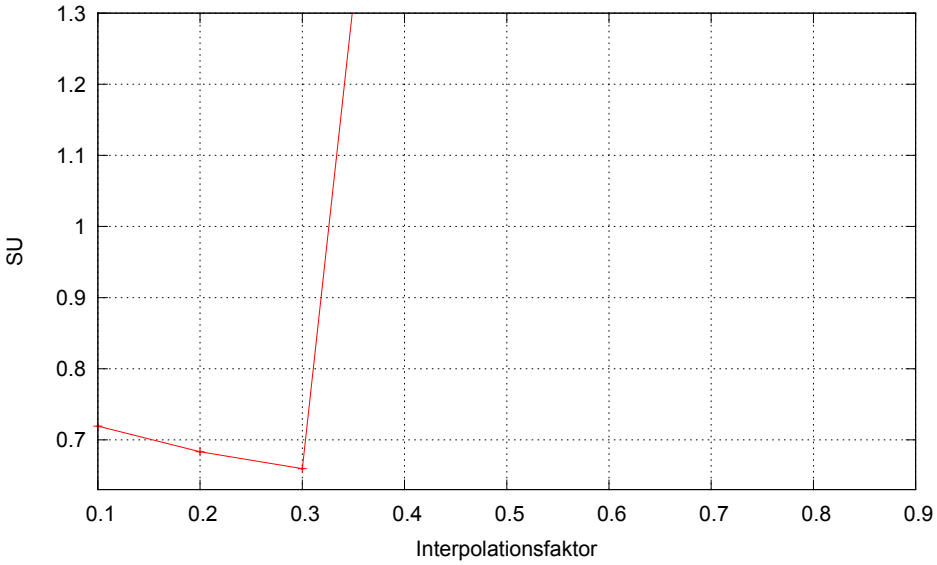

Abbildung 6.17: SU-Ergebnisse der Interpolation

Tabelle 6.17: Fehlerraten-Ergebnisse der Interpolation
In den Spalten steht der Faktor der Interpolation. Die Werte in den Zellen sind in Prozent angegeben.

	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
SER	93,85	**93,63**	97,02	229,06	233,15	287,60	284,86	283,01	283,07
SU	71,94	68,33	**65,95**	197,16	201,93	266,63	267,35	267,55	268,01

Die beste Sentence-Unit-Errorrate ergibt sich bei einem Interpolationsfaktor von 0,3. Die beste Slot-Errorrate dagegen entsteht bei einem Faktor von 0,2. Dies deutet darauf hin, dass bei einem Faktor von 0,3 mehr Satzbegrenzungen erkannt werden. Jedoch werden mehr Satzzeichenverwechslungen begangen und weniger Zeichen korrekt gesetzt.
Bei allem Tests erweist sich die Verwechslungsrate als sehr hoch. Dies zeigt sich an den großen Differenzen zwischen SER und SU-Errorrate.
Außerdem lässt sich auch hier wieder ab einem Interpolationsgewicht von 0,4 ein starker Anstieg der Fehlerwahrscheinlichkeiten erkennen, da dann dem stark umgewichteten Entscheidungsbaum höheres Gewicht als dem Sprachmodell gegeben wird.

6.3.4 Analyse

Die Heruntergewichtung der Wahrscheinlichkeiten der Klasse „NONE“ führt zu deutlich weniger Auslassungsfehlern und damit zu höheren Recall-Raten als bei den beiden einzelnen Modellen. Beim Betrachten der Tabellen gibt es noch folgende Auffälligkeiten: Der Unterschied zwischen Precision und SU-Precision ist sehr groß. Dies gilt auch für die anderen Bewertungsmaßstäbe. Des Weiteren sind die Fehlerraten relativ hoch.

6.3.4.1 Hohe Verwechslungsraten

Die starken Unterschiede der Precision-, Recall- und Fehlerraten bei Satzgrenzen- und Zeichenbetrachtung weisen darauf hin, dass zwar viele Satzgrenzen richtig erkannt werden, dann aber die falsche Zeichen gesetzt werden.

Ein Test zeigt, dass 1606 von 4408 gesetzten Zeichen (36,43 %) Verwechslungen darstellen. In 383 dieser Fälle wird ein Komma statt eines Punktes gesetzt, in 1223 Fällen ein Punkt statt eines Kommas.

6.3.4.2 Fehler durch Spracherkennerfehler

Wird das kombinierte System nicht auf die Spracherkennerausgaben angewandt sondern auf die Referenztexte selbst, ergeben sich niedrigere Fehlerraten, wie es in Tabelle 6.18 erkennbar ist.

Tabelle 6.18: Fehlerraten des Systems auf dem Referenztext
Die Werte in den Zellen sind in Prozent angegeben.

Bewertungsmaßstab	auf Hypothesentext	auf Referenztext
Precision	33,53	55,34
SU-Precision	69,96	79,41
Recall	28,59	50,97
SU-Recall	59,66	73,14
SER	97,02	68,00
SU-Fehlerrate	65,95	45,83

Die Zahl der Verwechslungen gegenüber den Tests auf Spracherkennerausgaben sinkt um etwa 25 % und die Zahl der korrekt erkannten Zeichen steigt um etwa 87 %. Dadurch sinken die Fehlerraten. Des Weiteren sinkt die Zahl der nicht erkannten Zeichen (Deletions), wodurch der Recall steigt. Er nähert sich sogar ziemlich gut der Precision an.

6.3.4.3 Beispielausgaben

Die Zitate in Tabelle 6.19 sind den Development-Daten entnommen. Es wird nur eine Referenz aufgeführt, da sich die Referenzen bei diesem Beispiel nicht unterscheiden. Der Punkt in den Ausgaben dient als Satzendezeichen, theoretisch könnte auch ein Fragezeichen korrekt sein.
Die Zitate zeigen im Vergleich der Hypothesen mit der Referenz, dass es dem System gelingt, den Wortfluss in Satzeinheiten zu strukturieren. Beim Vergleich zwischen den Ausgaben des Systems auf den korrekten Referenzworten mit den Ausgaben des Systems auf den Hypothesenworten wird außerdem deutlich, dass viele Fehler durch Spracherkennerfehler zustande kommen.

Tabelle 6.19: Beispielausgaben des Systems
R: Referenz, HR: Hypothese auf Referenzworten, HH: Hypothese auf Hypothesenworten

R	And	why	are	you	saying	that	Gerry	Adams	should	be	questioned	.
HR	And	why	are	you	saying	that	Gerry	Adams	should	be	questioned	.
HH	And	why	are	you	saying	that	Gerry	Adams	should	be	questioned	.
R	Well	,	everyone	knows	.	Gerry	Adams	denies	he	was	ever	a
HR	Well	,	everyone	knows	,	Gerry	Adams	denies	he	was	ever	a
HH	Well	.	Everyone	knows	that	even	Terry	denies				
R	member	of	the	IRA	,	but	everyone	knows	that	he's	talking	
HR	member	of	the	IRA	,	but	everyone	knows	that	he's	talking	
HH	number	of	their	it		that	everyone	knows		is	talking	
R	rubbish	.		He	should	be	brought	in	for	questioning		along
HR	rubbish	.		He	should	be	brought	in	for	questioning		along
HH	rubbish	,	and	he	should	be	president		for	question	.	Along
R	with		Dolores	Price	.							
HR	with		Dolores	Price	.							
HH	with	the	lowest	price	,							

6.4 Test mit Unterscheidung zwischen Punkt und Fragezeichen

Die ermittelte beste Kombination aus Sprachmodell und Entscheidungsbaum wird, wie im Entwurfskapitel 4.1 auf S. 23 beschrieben, auch mit der Unterscheidung zwischen Punkt und Fragezeichen getestet. Bezüglich des Satzzeichens Fragezeichen ergeben sich dabei die in Tabelle 6.20 aufgezeigten Ergebnisse.
Da es sich um die Ergebnisse eines einzelnen Zeichens handelt, werden an dieser Stelle keine Sentence-Unit-Ergebnisse sondern konkrete Zeichenergebnisse betrachtet.

Tabelle 6.20: Ergebnisse des Fragezeichentests
Die Werte in den Zellen sind in Prozent angegeben.

Precision	7,77
Recall	8,28
SER	122,49

Im Gesamtsystem, das Sprachmodell und Entscheidungsbaum interpoliert und außerdem zwischen Kommata, Punkten und Fragezeichen unterscheidet, ergeben sich damit bezüglich aller Zeichen die in Tabelle 6.21 aufgelisteten Ergebnisse.

Tabelle 6.21: Ergebnisse des Gesamtsystems
Die Werte in den Zellen sind in Prozent angegeben.

	ohne Fragezeichen	mit Fragezeichen
Precision	33,53	28,88
SU-Precision	69,96	69,96
Recall	28,59	24,63
SU-Recall	59,66	59,66
SER	97,02	100,99
SU-Fehlerrate	65,95	65,95

Man erkennt, dass gegenüber den Ergebnissen ohne Unterscheidung zwischen Punkt und Fragezeichen, die Slot-Errorrate (SER) um 3,5 % und die SU-Fehlerrate um 0,8

% steigt. Sowohl die Precision- als auch die Recallwerte sinken. Dies liegt an den sehr niedrigen Precision- und Recallwerten der Fragezeichenerkennung. Sie bringt eine hohe Verwechslungsrate mit sich.

6.4.1 Analyse: Gründe für Verwechslungen von Punkt und Fragezeichen

Durch den Entschluss für ein Fragezeichen aufgrund eines Fragewortes am Satzanfang ergeben sich zwar in den meisten Fällen korrekte Entscheidungen allerdings auch Restriktionen. So werden beispielsweise Nebensätze, die mit einem Relativ- oder Interrogativpronomen beginnen und am Anfang eines Satzes stehen, mit Fragesätzen verwechselt. Zum Beispiel würde der Satz „What is good for you is good for me“ fälschlicherweise mit einem Fragezeichen versehen.
Des Weiteren ergeben sich Folgefehler durch vorher falsch gesetzte Fragezeichen. Wurde beispielsweise vor einem Relativ- oder Interrogativpronomen ein Punkt statt eines Kommas gesetzt, wird das Pronomen als Fragewort interpretiert und der folgende Satz mit einem Fragezeichen beendet. Tabelle 6.22 zeigt ein Beispiel aus den verwendeten Development-Daten.

Tabelle 6.22: Beispiel für Punkt-Fragezeichen-Verwechslung

Referenz	It	was	in	an	earlier	book	by	the	journalist	Ed
Hypothese	It	was	in	an	earlier	book	by	the	journalist	said
Referenz	Moloney	,	who	is	publishing	this	new	book	,	
Hypothese	Malone		whose		publishing	this	new	book	.	
Referenz	who	made	that	allegation	in	his	book	.		
Hypothese	Who	made	that	allegation	in	his	book	?		

Ein ähnlicher Fehler entsteht, wenn das Modell fälschlicherweise einen Punkt vor einem Verb setzt, obwohl der Satz noch nicht zu Ende war. Dann wird das Verb zum ersten Wort im nächsten Satz und der Satz mit einem Fragezeichen versehen. Tabelle 6.23 zeigt ein Beispiel aus den verwendeten Development-Daten.

Tabelle 6.23: Weiteres Beispiel für Punkt-Fragezeichen-Verwechslung

Referenz	Well	,	there	again	is	that	there	is	a	legal	system
Hypothese	Well		there	again	.	Is	that	there	is	a	legal	system
Referenz	and	process	that	says	that	anyone	who	commits	an		
Hypothese	and	process	that	says	that	anyone	who	commits	an		
Referenz	act	that's	of	a	criminal	nature	should	be	punished	.	
Hypothese	act	that	of	a	criminal	nature	should	be	punished	?	

Eine umgekehrte Verwechslung, also dass fälschlicherweise ein Punkt statt eines Fragezeichens gesetzt wird, geschieht in Fällen der Herauszögerung des Fragewortes durch eine vorangestellte Konjugation. Beispielsweise würde das System bei vorliegendem Satz „But what do you think about it?“ ein Punkt statt eines Fragezeichens setzen.

6.4.1.1 Punkt-Fragezeichen-Verwechslung auf Referenztexten

Nimmt man als Eingaben für die Systeme Referenztexte mit gegebenen Satzanfängen, verringert sich der Fragezeichenerkennungsansatz auf das Problem, bei einem gegebenen Punkt zu entscheiden, ob der Punkt gerechtfertigt ist oder ob an diese Stelle eher ein Fragezeichen zu setzen ist. Der vorgestellte Ansatz liefert bei einem Test eine Precision von 53,47 % und einen Recall von 31,69%. Die Schwächen des Systems liegen wie erwähnt in Nebensätzen, die zu Beginn eines Satzes stehen. Dort werden fälschlicherweise Fragesätze vermutet. Als Beispiel aus den Developement-Daten diene der folgende Satz.

> What we found in pre-season tournaments was that you had a lot less yellow cards because players didn't want to stay on the sidelines for ten.

Deletion-Fehler entstehen hauptsächlich an Stellen, an denen dem Fragewort eine Konjugation wie „and“, „but“ oder „so“ vorausgeht. Da als mögliches Fragewort nur das erste Wort eines Satzes betrachtet wird, werden diese Fälle nicht als Fragesätze erkannt. Ein Beispielsatz aus den Development-Daten verdeutlicht dies:

> So you will never put it behind you until you know?

Eine Verbesserung des Systems wäre möglich, wenn bei bestimmten Schlüsselwörtern mehr als nur ein Wort am Satzanfang überprüft wird.

6.5 Analyse: Vergleich von Sprachmodell, Entscheidungsbaum und Endsystem

Im Balkendiagramm 6.18 sind links die zu maximierenden und rechts die zu minimierenden Bewertungsmaßstäbe aller verwendeten Modelle dargestellt.

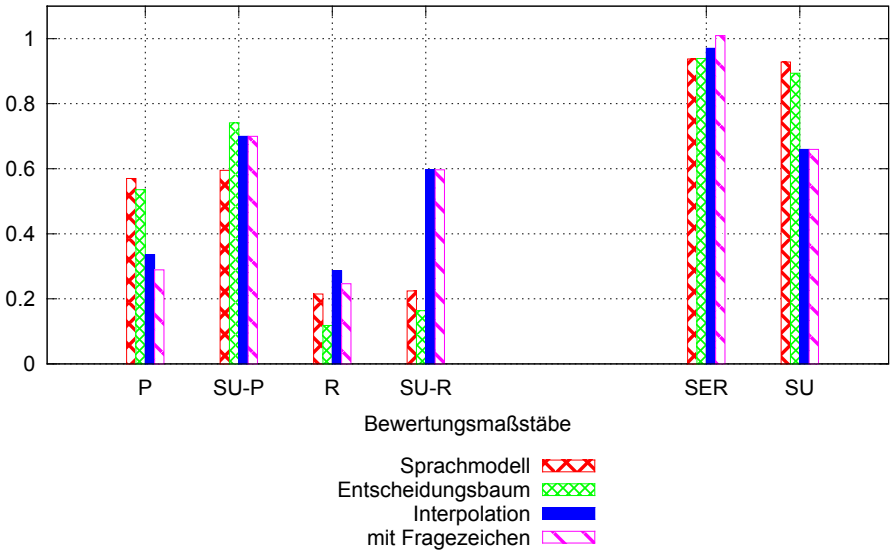

Abbildung 6.18: Ergebnisse der Systeme im Vergleich

Ein Vergleich von Sprachmodell und Entscheidungsbaum ergibt, dass das Sprachmodell fast überall leicht besser abschneidet. Nur bei den Maßstäben SU-Precision und SU-Fehlerrate liefert der Entscheidungsbaum bessere Ergebnisse. Daraus kann man schließen, dass die Part-of-speech-Tags und prosodischen Merkmale, die dieser betrachtet, ebenso wichtig für die Erkennung von Satzbegrenzungen sind wie die textuellen Merkmale, die das Sprachmodell verwendet. Allerdings liefern die konkreten Wörter präzisere Ergebnisse hinsichtlich der Wahl der korrekten Satzzeichen. Tendenziell setzt das Sprachmodell auch mehr Zeichen als der Entscheidungsbaum, wodurch die Recallwerte steigen.
Ein Grund für das schlechtere Abschneiden des Entscheidungsbaums gegenüber dem Sprachmodell könnte an der sehr viel geringeren Trainingsdatenzahl liegen.
Dass beide Systeme Schwäche besitzen, kann man an den Fehlerraten klar erkennen. Beide Systeme haben einen sehr niedrigen Recall, setzen tendenziell also viel zu wenige Zeichen. Dennoch kann die Kombination beider Systeme einen Teil ihrer Schwächen beheben: Bei den Tests schneidet die Kombination aus Sprachmodell und Entscheidungsbaum am besten ab.
Die Berücksichtigung des Fragezeichens verschlechtert die Ergebnisse der Systeme ein wenig, da sich die Fragezeichenerkennung als sehr anfällig für Folgefehler herausstellt.

6.6 Vergleich mit verwandten Arbeiten

Ebenso wie in der Arbeit von Kim [6] wird ein Entscheidungsbaum auf prosodischen Merkmalen sowie Wortarten trainiert und mit einem hidden-N-Gramm-Sprachmodell interpoliert. Im Gegensatz zu der vergangenen Arbeit wird hier jedoch die Interpolation noch um eine Wahrscheinlichkeitenanpassung erweitert.
Gotoh [19] verwendet ebenfalls die Modelle Entscheidungsbaum und N-Gramm und erreicht ähnliche Satzgrenzen-(SU)-Fehlerraten wie die in diesem Kapitel vorgestellten. Allerdings interpoliert er die Wahrscheinlichkeiten nicht sondern nutzt die in Kapitel 2.5 vorgestellten Formeln.
Im Unterschied zu Kim und Gotoh wird nicht nur die reine Satzgrenzenerkennung

sondern auch die konkrete Unterscheidung zwischen den Satzzeichen getestet. Anders als in den vergangenen Arbeiten, die ebenfalls zwischen Satzzeichen differenzieren [16, 17, 20] und auch Fragezeichen betrachten [16, 20], wird die Entscheidung zwischen Punkt und Fragezeichen nicht von den Modellen selbst vorgenommen sondern auf einer höheren logischen Ebene. Huang verwendet Maximum-Entropie-Modelle für die Fragezeichenerkennung. [16] Vergleichbarer bezüglich der Fragezeichentests dieser Arbeit ist daher die Arbeit von Gravano. Dieser beschreibt, dass die schlechteren Ergebnisse von Fragezeichen gegenüber Punkten oder Kommata an der zu geringen N-Gramm-Ordnung liegen. [20] Trotz des Versuches, dieses Problem durch die beschriebene andere Herangehensweise zu umgehen, führt diese Arbeit bei der Fragezeichenerkennung zu schlechten Ergebnissen.
Ebenso wie [7] und [9] zeigt auch das hier erstellte und evaluierte System, dass die Satzgrenzenerkennung auf Referenztexten besser ist als auf Hypothesentexten des Spracherkenners. Dies liegt an Folgefehlern durch Fehleinschätzungen des Spracherkenners.

6.7 Zusammenfassung

Die Testergebnisse zeigen, dass das im Entwurf vorgestellte System mit der Formel

$$P(i) = 0,3 \cdot P_{Entscheidungsbaum}(i) + 0,7 \cdot P_{Sprachmodell}(i) \text{ (vgl. 4.5)}$$

die besten Ergebnisse liefert. Da die Modelle zu wenige Zeichen setzen (erkennbar an den hohen Recall-Werten), müssen ihre Wahrscheinlichkeiten für die Klasse „NONE“ niedriger und die Wahrscheinlichkeiten für die anderen Klassen entsprechend höher gewichtet werden. Bei dem Sprachmodell liefert eine konstante Umgewichtung mit Faktor 0,8 die besten Ergebnisse, während der Entscheidungsbaum abhängig von der Anzahl der Worte, die seit dem letzten Satzende gesehen wurden, umgewichtet werden sollte. Die folgenden Parameter liefern dabei die besten Ergebnisse:

$$\alpha_{Entscheidungsbaum} = 0,6 \cdot k + 0,4 \text{ (vgl. 4.1)}$$

Der verwendete Entscheidungsbaum wird mit den 125 am häufigsten vor oder nach einem Satzzeichen auftretenden Referenzwörtern sowie einem Kontext von 3 in die Vergangenheit und 1 in die Zukunft trainiert.
Eine Analyse zeigt, dass das Zeichensetzungssystem sehr oft Punkte statt Kommatas setzt.
Die Unterscheidung zwischen Punkt und Fragezeichen liefert trotz des neuen Ansatzes eher negative Ergebnisse. Diese kommen vor allem durch Folgefehler des Spracherkenners oder des Zeichensetzungssystems zustande.

7. Fazit und Ausblick

Das in dieser Arbeit erstellte System dient der automatischen Strukturierung von Spracherkennerausgaben. Dazu werden die Zeichen Punkt, Komma und Fragezeichen betrachtet. Das Zeichensetzungssystem arbeitet auf Basis der Wahrscheinlichkeitsschätzungen eines Entscheidungsbaums, der mit Wörtern, Wortarten und prosodischen Merkmalen trainiert wurde, und eines 4-Gramm-Sprachmodells, das nur Wörter betrachtet. Die Kombination aus beiden erweist sich als besser als beide Systeme einzeln. Zudem führt eine Umgewichtung der Wahrscheinlichkeiten, die dafür sorgt, dass diejenige für Satzzeichen höher gewichtet werden, zu einer merklichen Verbesserung der Fehlerrate.
Im direkten Vergleich schneidet der Entscheidungsbaum bezüglich der Satzgrenzenerkennung zwar besser ab, bezüglich der Bestimmung des korrekten Satzzeichens jedoch liefert das Sprachmodell bessere Ergebnisse.
Der Entscheidungsbaum weist eine SU-Fehlerrate von 89,36 % auf, das Sprachmodell eine SU-Fehlerrate von 92,82 %. Die Kombination beider Modelle nach vorheriger Umgewichtung der Wahrscheinlichkeiten führt zu einem besten SU-Fehlerratenwert von 65.95 % auf den Hypothesen des Spracherkenners und zu einem Wert von 45,83 % auf den Referenztexten.

Ein Grund für die hohe Auslassungsrate von Satzzeichen könnte in mangelnden Trainingsdaten liegen. Sowohl Sprachmodell als auch Entscheidungsbaum können durch weiteres Trainingsmaterial verbessert werden. Bei der Auswahl des Trainingsmaterials sollte allerdings darauf geachtet werden, dass die Zeichensetzung konsistent ist. Quaero bemüht sich durch die Erstellung von Richtlinien, eine konsistente Zeichensetzung zu erreichen. Allerdings enthalten die Trainingsdaten für das erstellte System auch Texte, die diesen Richtlinien nicht entsprechen. Als Beispiel seien die Zeitungsartikel genannt, mit denen das Sprachmodell trainiert wurde. Inkonsistente Zeichensetzung in den Trainingstexten führt aber dazu, dass die Modelle nicht robust trainiert werden können. Somit können sie auch keine verlässlichen Aussagen treffen, wenn sie getestet werden.
Von den Satzzeichen abgesehen erweist es sich für Trainings- und Testtexte auch auf Wortebene wichtig, dass diese normalisiert vorliegen. An dieser Stelle bleiben Verbesserungsmöglichkeiten: So wurde zwar Groß- und Kleinschreibung normalisiert

und alle Modelle mit kleinen Buchstaben trainiert, allerdings unterscheidet sich die Verwendung von Apostrophen zum Zusammenziehen zweier Worte in Trainings- und Testtexten teilweise. Beispiele hierfür stellen Vorkommen von „he's“ und „he is“ oder „what's“ und „what is“ dar. Eine vollständige Normalisierung aller verwendeten Texte kann die Fehlerrate wahrscheinlich ein wenig senken.
Eine Möglichkeit, den Entscheidungsbaum zu verbessern, kann außerdem durch die Hinzunahme weiterer prosodischer Merkmale erfolgen. So zeigen viele Arbeiten auf diesem Gebiet, dass der Verlauf der Sprechfrequenz neben der Pause ein sehr verlässliches Merkmal für das Erkennen von Satzgrenzen darstellt.
Des Weiteren bilden N-Gramme und Entscheidungsbäume nicht die einzigen Modelle, die für den Zweck der Zeichensetzung verwendet werden können. Aus diesem Grund wurden im Grundlagenkapitel dieser Arbeit auch andere Modelle vorgestellt. So wie Entscheidungsbaum und N-Gramm bei ihrer Kombination gegenseitig Schwächen ausgleichen, könnte auch die Hinzunahme weiterer Modelle die Fehlerraten weiter senken.

Literaturverzeichnis

[1] Lynne Truss. *Eats, Shoots and Leaves.* 2003.

[2] Evgeny Matusov, Dustin Hillard, Mathew Magimai-Doss, Dikel Hakkani-Tur, Mari Ostendorf, and Hermann Ney. Improving speech translation with automatic boundary prediction. *Proceedings of Interspeech 2007*, 2007.

[3] Sadaoki Furui, Tomonori Kikuchi, Yousuke Shinnaka, and Chiori Hori. Speech-to-text and speech-to-speech summarization of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 2004.

[4] Christian Fügen, Muntsin Kolss, Matthias Paulik, Sebastian Stüker, Tanja Schultz, and Alex Waibel. Open domain speech translation: From seminars and speeches to lectures. *TC-STAR workshop on speech to speech translation, Barcelona, Spain, pp.81-86*, 2006.

[5] Alexander Waibel. Vorlesung kognitive systeme, 2010.

[6] Joungbum Kim, Sarah E. Schwarm, and Mari Ostendorf. Detecting structural metadata with decision trees and transformation-based learning. *Proceedings of North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, 2004.

[7] Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transaction on Audio, Speech and Language Processing, vol. 14, no. 5*, 2005.

[8] Elisabeth Selkirk. Sentence prosody: Intonation, stress, and phrasing. *Handbook of phonological theory*, 1995.

[9] Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Mari Ostendorf, Dilek Hakkani, Madelaine Plauche, Gökhan Tür, and Yu Lu. Automatic detection of sentence boundaries and disfluencies based on recognized words. *Proceedings of the International Conference on Spoken Language Processing, vol. 5*, 1989.

[10] Adam L. Berger, Stephen A. Della Pietra, and Vincent A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 1996.

[11] Tom M. Mitchell. *Machine Learning.* 1997.

[12] Johne Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of International Conference on Machine Learning 2001*, 2001.

[13] William Morgan. Sentence unit detection without an audio signal. 2009.

[14] Hanna M. Wallach. Conditional random fields: An introduction. *University of Pennsylvania CIS Technical Report MS-CIS-04-21*, 2004.

[15] Wei Lu and Hwee Tou Ng. Better punctuation prediction with dynamic conditional random fields. *Proceedings of Conference on Empirical Methods on Natural Language Processing*, 2010.

[16] Jing Huang and Geoggrey Zweig. Maximum entropy model for punctuation annotation from speech. *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[17] Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. Punctuation annotation using statistical prosody models. *Proceedings of the International Speech Communication Association Workshop on Prosody in Speech Recognition and Understanding*, 2001.

[18] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. Performance measures for information extraction. *Proceedings of the DARPA BN Workshop*, 1999.

[19] Yoshihiko Gotoh and Steve Renals. Sentence boundary detection in broadcast speech transcripts. *Proceedings of International Speech Communication Association Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000*, 2000.

[20] Agustín Gravano, Martin Jansche, and Michiel Bacchiani. Restoring punctuation and capitalization in transcribed speech. *International Conference on Acoustics, Speech, and Signal Processing 2009*, 2009.

[21] Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communications, vol. 32, no. 1-2*, 2000.

[22] Don Baron, Elizabeth Shriberg, and Andreas Stolcke. Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. *Proceedings of the International Conference on Spoken Language Processing, vol. 2*, 2002.

[23] Elizabeth Shriberg, Andreas Stolcke, and Don Baron. Can prosody aid the automatic processing of multi-party meetings? evidence from predicting punctuation, disfluencies and overlapping speech. *Proceedings of International Speech Communication Association Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, 2001.

[24] http://www.deutschseite.de/vokabeln/fragewoerter/fragewoerter.pdf.

[25] Andreas Stolcke. Srilm - an extensible language modeling toolkit. *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[26] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Technical Report TR-10-98, Computer Science Group, Harvard University*, 1998.

[27] http://www.ims.uni-stuttgart.de/projekte/corplex/treetagger/decisiontreetagger.html.

[28] http://www.ims.uni-stuttgart.de/projekte/corplex/treetagger/penn-treebank-tagset.pdf.

[29] Jáchym Kolář and Lori Lamel. On development of consistently punctuated speech corpora. 2011.

A. Frage- und Schlüsselwörter

A.1 Verwendete Fragewörter

when	who	wherefrom	which
why	how	whereto	whose
what	where	whom	

A.2 Die 125 absolut am häufigsten auftretenden Schlüsselwörter vor und nach Satzzeichen

the	as	time	two
and	or	one	her
said	his	there	however
in	was	all	s
a	year	percent	your
of	you	more	us
to	be	including	have
he	by	about	march
on	from	if	then
but	they	wrote	here
it	not	them	like
this	are	comment	united
for	years	up	do
i	she	now	when
is	their	were	says
point	we	no	too
with	an	has	him
at	new	people	out
which	so	my	states
who	last	other	june
that	its	may	week

where	april	president	iraq
july	million	country	state
world	me	well	right
day	tuesday	next	down
according	wednesday	than	war
our	friday	since	nov
news	first	oct	points
while	thursday	ago	china
will	monday	york	
some	after	been	
government	good	sept	

A.3 Schlüsselwörter aus den Quaero-Richtlinien extrahiert

and	if	that	why
but	since	who	what
for	when	which	how
or	while	right	much
nor	to	okay	many
so	before	hello	where
after	yes	good	whom
although	however	morning	whose
as	well	evening	
because	too	when	

B. Die vollständigen Evaluierungsergebnisse

Die vollständigen Evaluierungsergebnisse befinden sich auf der beiliegenden CD.