Gefördert vom
bmb+f
Bundesministerium für
Bildung, Wissenschaft,
Forschung und Technologie

Verbmobil
Verbundvorhaben

Codebuchübergreifende Bucket-Box-Intersection zur schnellen Berechnung von Emissionswahrscheinlichkeiten im Karlsruher VM-Erkenner

Monika Woszczyna
Jürgen Fritsch

TU Karlsruhe

Vm

Report Nr 211
Juli 1997

Juli 1997

Monika Woszczyna
Jürgen Fritsch

Lehrstuhl Prof. Waibel
Institut für Logik, Komplexität und Deduktionssysteme
Fakultät für Informatik
Universität Karlsruhe
Am Fasanengarten 5
76131 Karlsruhe

Tel.: (0721) 608 - 4732
e-mail: {monika|fritsch}@ira.uka.de

Gehört zum Antragsabschnitt: 1.4.1 Integration des Multilingualen Erkenners

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 G gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

Inhaltsverzeichnis

1 Einleitung

Für die Integration des Verbmobil-Prototypen im Juli 1997 wurden verschiedene Techniken zur Beschleunigung der Erkennung von uns zum ersten Mal auf dem deutschen Terminabsprache-Task im Verbmobil-Szenario (VM-Deutsch) eingesetzt.

In diesem Dokument werden insbesondere Verfahren zur schnellen Berechnung der HMM Emissionswahrscheinlichkeiten näher beschrieben und ihr Nutzen auf dem deutschen Terminabsprache-Task (VM-Deutsch) mit den Ergebnissen früherer Experimente auf dem *North-American-Business-News-Task* (NAB) verglichen. Diese Verfahren werden von uns inzwischen auch für die im Rahmen von Verbmobil entwickelten Japanischen und Englischen Erkenner eingesetzt.

Sowohl das *Bucket-Box-Intersection* Verfahren (BBI) alsauch unsere neuere Variante davon, das codebuchübergreifende *Big-BBI* Verfahren (B-BBI) bringen auch bei der Anwendung auf die VM-Szenarios erhebliche Verbesserungen. Da diese Verfahren einen Kompromiß zwischen Wortfehlerrate und Geschwindigkeit anstreben, funktionieren sie allerdings um so besser, je kleiner die Ausgangsfehlerrate ist.

2 BBI und B-BBI

2.1 Bucket Box Intersection (BBI)

Das *Bucket-Box-Intersection* Verfahren [Fritsch und Rogina, 1996] ist eine Erweiterung des *Bucket-Voronoi* Verfahrens [Ramasubramanian und Paliwal, 1992], [Fritsch et al., 1995]. Mittels des *Bucket- Box* Verfahrens können schnell die Verteilungen eines Codebuchs gefunden werden, die den geringsten Abstand zum Eingabevektor haben.

Dazu wird um die Gaußverteilung jedes Codebuchvektors eine Schachtel gelegt, die die Gaußglocke bei einem Grenzwert T abschneidet. Für die Berechnung der Bewertungen werden nur die Verteilungen verwendet, in deren Schachtel der Eingabevektor liegt. Der Maximale Fehler dieser Methode ist durch T gegeben.

Um mit niedrigem Aufwand herauszufinden, in welche Schachteln ein Eingabevektor fällt, wird ein Fragenbaum aufgebaut: Ist der Koeffizient eines Eingabevektors kleiner als der Achsenabschnitt der durch die Frage an einer Verzweigung des Baumes definierten Trennebene im Merkmalsraum, so fällt er in eine der Schachteln links von der Verzweigung, sonst in eine Schachtel rechts von der Verzweigung.

Der BBI Baum (einer für jedes Codebuch) wird nach dem Training des Systems einmal vorberechnet. Dazu werden entlang aller Achsen im Raum zunächst die Codebuchvektoren die noch in dem zu Teilenden Knoten liegen sortiert. Dann wird eine Trennebene für jede Achse so gewählt, daß auf jeder Seite der Trennebene

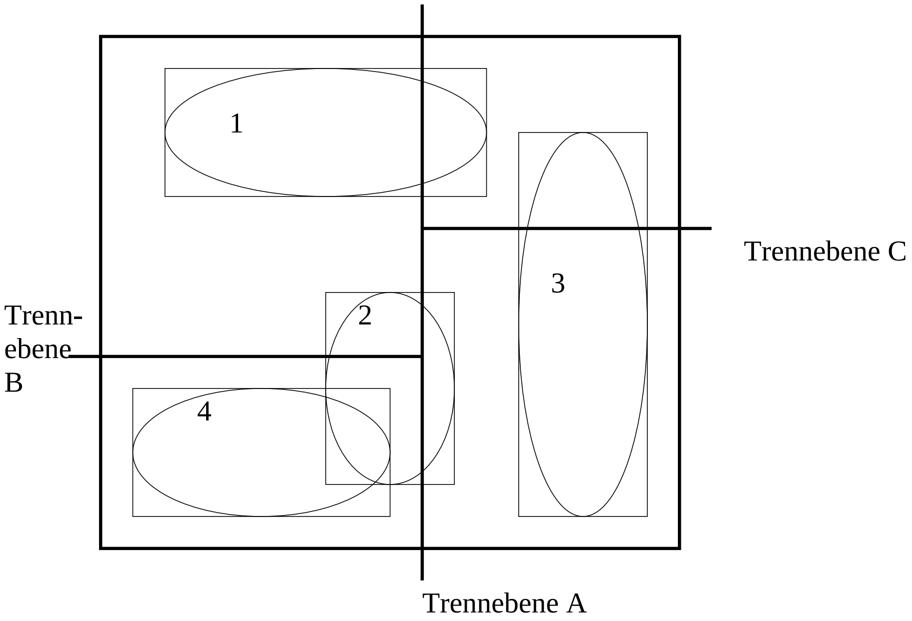

Abbildung 1: Bucket Box Intersection im zweidimensionalen Raum.

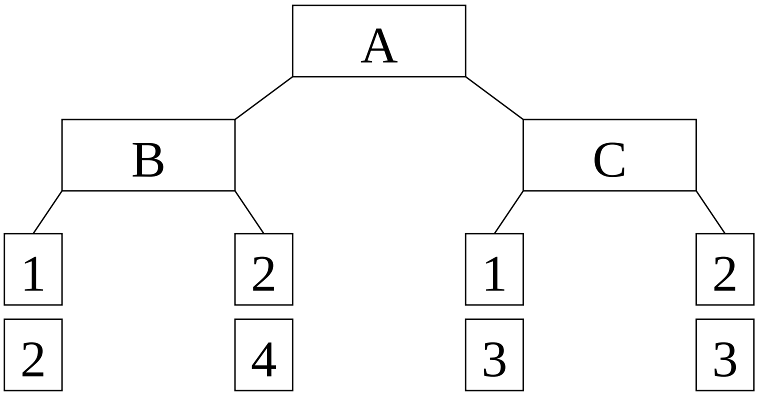

Abbildung 2: Bucket Box Intersection Baum.

gleich viele Vektoren liegen. Für die nächste Unterteilung des Baums wird die Trennebene derjenigen Achse gewählt, die die wenigsten der zu den Codebüchern gehörenden Schachteln schneidet. Codebuchvektoren, deren Schachteln durch die so gefundene Trennebene dennoch geschnitten werden, kommen in beide der Nachfolge-Knoten. Die Endknoten, die jeweils mehrere Codebuchvektoren enthalten, werden auch *Buckets* (Eimer) genannt.

Dieses Verfahren produziert zwar kein optimales Ergebnis, erzeugt aber effizient einen gut balancierten Baum.

Das BBI Verfahren gibt die beste Beschleunigung für Systeme mit vergleichsweise wenigen, großen Codebüchern, ist also insbesondere für semikontinuierliche Systeme geeignet.

2.2 Codebuchübergreifendes BBI (B-BBI)

Im Karlsruher Erkenner werden wegen der besseren Erkennungsleistung oft voll kontinuierliche Systeme mit mehreren tausend kleinen Codebüchern verwendet, für die sich das *Bucket-Box* Verfahren nicht besonders gut eignet.

Wir rechnen deshalb einen großen BBI-Baum über die Menke der Vektoren aller Codebücher, um so die Vektoren auszuwählen, mit denen die Bewertungen berechnet werden sollen. Wird eine Bewertung für ein weit vom Eingabevektor entferntes Codebuch angefragt, von dem kein Vektor im zu dem Eingabevektor gehörigen BBI-Bucket liegt, muß ein Backoff-Vektor gefunden werden, mit dem die Abstandsberechnung durchgeführt werden kann. Als erste Näherung haben wir zunächst den häufigsten Vektor des Codebuchs im Training benutzt, was sich aber als zu ungenau erwiesen hat.

Anstelle des im Trainings am häufigsten gesehenen Vektors kann auch bei der Erstellung des B-BBI-Baumes für jedes Bucket und jedes nicht im Bucket vertretene Codebuch der Vektor berechnet werden, der am nächsten an diesem Bucket liegt. Dabei wird der Abstand der Box des Vektors zum Rand des Buckets als Abstandsmaß verwendet.

3 Systembeschreibung

Für beide Systeme wurden vergleichbare reduzierte Varianten verwendet; allerdings basieren beide Systeme direkt auf Evaluationssystemen, d.h. es wurden nicht extra kleinere, auf Geschwindigkeit optimierte Systeme trainiert. Auf manche Möglichkeiten zur Beschleunigung der Systeme wurde in diesen Experimenten verzichtet, um die etwas älteren NAB-Testreihen noch mit den neueren VM-Testreihen vergleichen zu können.

3.1 Deutsche Terminabsprache (VM-Deutsch)

Das für die hier beschriebenen Experimente verwendete Basissystem ist eine abgespeckte Variante des für die VM-Evaluation 1996 trainierten Systems.

Für die Modellierung der Allophon-Zustände werden 2500 Codebücher mit je 32 Vektoren zu je 32 Koeffizienten aus den über 17000 beobachteten Polyphon-Zuständen gebildet. Das abgespeckte System ist voll-kontinuierlich. Dadurch wird das System bei moderatem Verlust an Genauigkeit kleiner und etwas schneller. Für die Berechnung der Bewertungen wird nur der Abstand des Merkmalsvektors zum nächsten Vektor in jedem Codebuch verwendet, was eine deutliche Zeitersparnis aber wiederum einen Verlust an Genauigkeit mit sich bringt.

Beim Erkennungsvorgang wird nur der erste und der letzte Suchdurchgang durchgeführt, was zu erheblichen Einbußen an Wortgenauigkeit führt. Da der mittlere Suchdurchgang allerdings gut 30% der Rechenzeit benötigt, ist der Verlust an Genauigkeit kleiner als bei anderen Methoden zur Reduktion der Rechenzeit.

Des Weiteren wurden für diese Experimente keine Adaption und keine Vokaltrakt-Längen-Normierung (VTLN) verwendet. Zudem werden die Pruning-Schwellen für dieses System deutlich kleiner gehalten als bei dem ursprünglichen Evaluationssystem.

Vokabular und Sprachmodell entsprechen dem des Pflicht-Systems der Evaluation. Im Vergleich zum ursprünglichen Evaluationssystem hat dieses System eine Fehlerrate von ca 17-18% auf den Daten der Evaluation von 1996; dabei liegt die Rechenzeit auf einer Sparc-Ultra Workstation bei ca 6-8 mal als Echtzeit.

3.2 North American Business News (NAB)

Für diese Experimente wurde ein (inzwischen etwas veraltetes) System mit 3000 Codebüchern verwendet. Jedes Codebuch enthält 32 Vektoren zu 48 Koeffizienten. Auch dieses System hat pro Codebuch nur eine Verteilung.

Wie beim Terminabsprache-System wurde bei der Berechnung der Emissions-Wahrscheinlichkeiten jeweils nur der nächste Codebuchvektor berücksichtigt. Auch hier wurde für die Experimente keine Adaption und VTLN eingesetzt. Dieses System erreicht eine Wortfehlerrate von minimal 9% auf dem Testset der NAB'94 Evaluation (das beste für den Karlsruher Erkenner verfügbare System hat im Vergleich dazu eine Fehlerrate von 7%).

Beim Erkennungsvorgang wurde wiederum nur der erste und letzte Suchdurchgang verwendet. Dadurch steigt die minimal erreichbare Fehlerrate auf 10.5%.

4 Experimente

4.1 Deutsche Terminabsprache (VM-Deutsch)

Die Experimente wurden auf dem vollen Evaluationsset (343 Aufnahmen) der VM-Evaluation 1996 durchgeführt. Auf eine Einzelbeurteilung der Fehler (Adjudication), wie sie in der Evaluation durchgeführt wurde, wurde allerdings verzichtet.

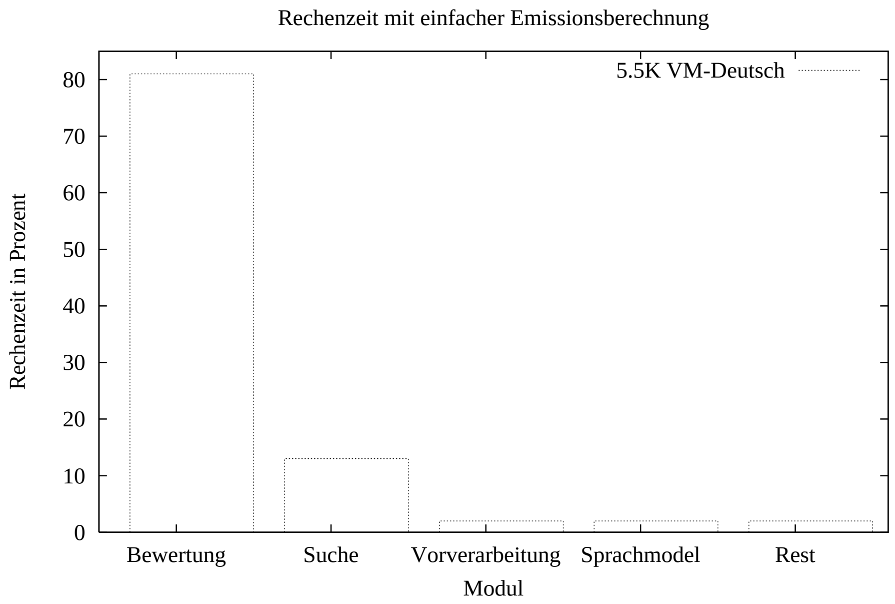

Abbildung 3: Rechenzeitverteilung bei einfacher Berechnung der Emissionswahrscheinlichkeiten für VM-Deutsch.

Für die Integration des Deutschen Erkenners in das Verbmobil System mußten zunächst die rechenzeitaufwendigen Komponenten in der Erkennung ermittelt werden. Abbildung 3 zeigt eine Analyse der Rechenzeitverteilung für das bereits leicht abgespeckte Evaluationssystem.

Wie man deutlich sieht, ist die Beschleunigung der Berechnung der Bewertungen, also der Emissionswahrscheinlichkeiten in den Hidden Markov Modellen der offensichtliche Angriffspunkt. Für den Englischen und Japanischen Verbmobil Erkenner stellt sich die Situation ähnlich dar; zur genauen Modellierung der Allophone werden nämlich in all diesen Systemen mehrere tausend kontextabhängige Codebücher eingesetzt, so daß die Ermittlung der Vektorabstände für alle Codebücher sehr aufwendig ist. Deshalb wurde die schon für NAB entwickelten BBI und B-BBI Techniken auch für die Verbmobil-Tasks eingesetzt.

Abbildung 4 zeigt, daß mit codebuchübergreifendem B-BBI die Gesamtrechenzeit für die Erkennung bei einer konstanten Fehlerrate von ca 20% von über 4 mal

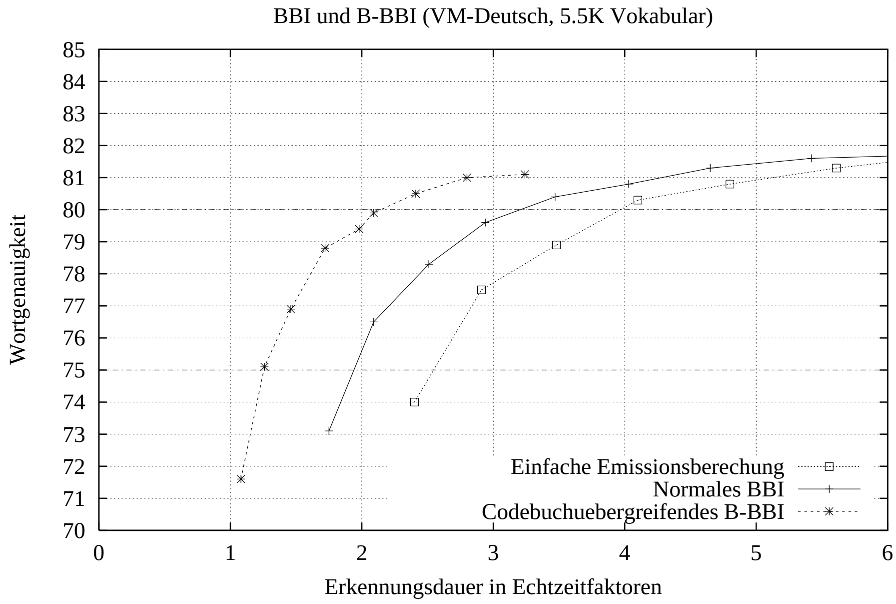

Abbildung 4: Bucket-Box-Intersection und codebuchübergreifende Big-Bucket-Box-Intersection für VM-Deutsch mit ca 5500 Wörtern im Vokabular.

langsamer als Echtzeit auf fast zwei mal Echtzeit gesenkt werden kann. Um die Rechengeschwindigkeit bei gleicher Wortgenauigkeit vergleichen zu können, wurden diese Experimente mit einer Reihe von verschiedenen Einstellungen der Pruning-Parameter in der Suche durchgeführt. Für die BBI-Reihe wurde ein Baum mit einer Tiefe von 4, für die BBBI-Reihe mit einer Tiefe von 8 verwendet. In beiden Fällen wurden die Gaußkverteilungen für die Bestimmung der "Schachteln" bei 50% ihres Maximalwertes abgeschnitten. Diese Werte wurden auf einer Kreuzvalidierungsmenge bestimmt. Kleinere Abweichungen vom Optimum (40% statt 50%, Tiefe 9 statt 8) beeinflussen das Gesamtergebnis aber nur wenig.

In Abbildung 5 sieht man, wie durch den Einsatz von B-BBI die Rechenzeitverteilung zwischen den Modulen beeinflußt wird.

4.2 North American Business News (NAB)

Wie man in Abbildung 6 sieht, wurde auch beim NAB-Task mit 64000 Wörtern im Vokabular der Hauptanteil der Rechenzeit in der Erkennung für die Berechnung der Emissionswahrscheinlichkeiten verbraucht. Bei kleineren Vokabularen, (etwa vergleichbar mit den für die VM-Szenarions verwendeten) ist diese Verteilung sogar noch ausgeprägter. Ein Ansatz zur Reduzierung dieses Aufwandes erscheint demnach zur Beschleunigung des Systems sinnvoll.

Abbildung 7 zeigt die deutliche Reduktion der Gesamtrechenzeit für eine Wortgenauigkeit von 85% von 6.9 mal langsamer als Echtzeit auf 3.7 mal langsamer als

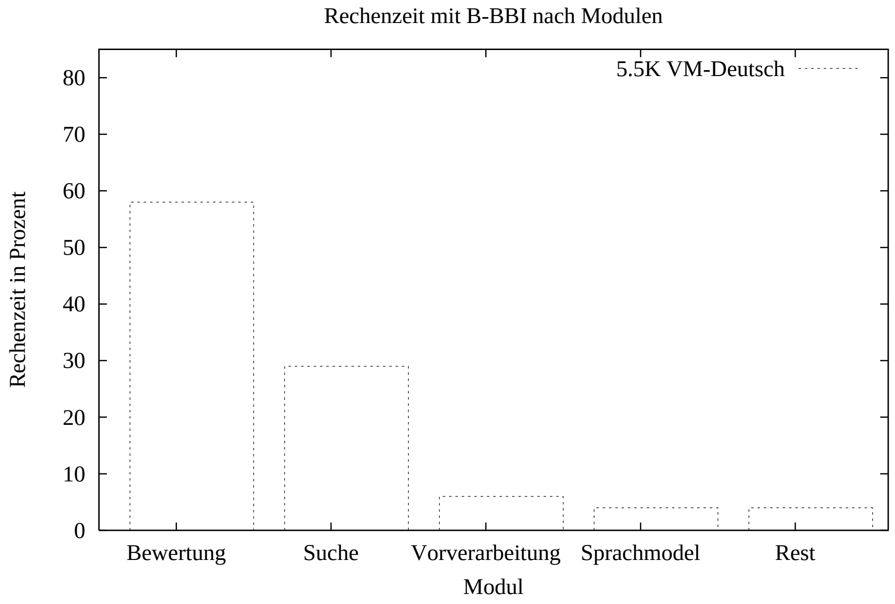

Abbildung 5: Rechenzeitverteilung mit B-BBI für VM-Deutsch

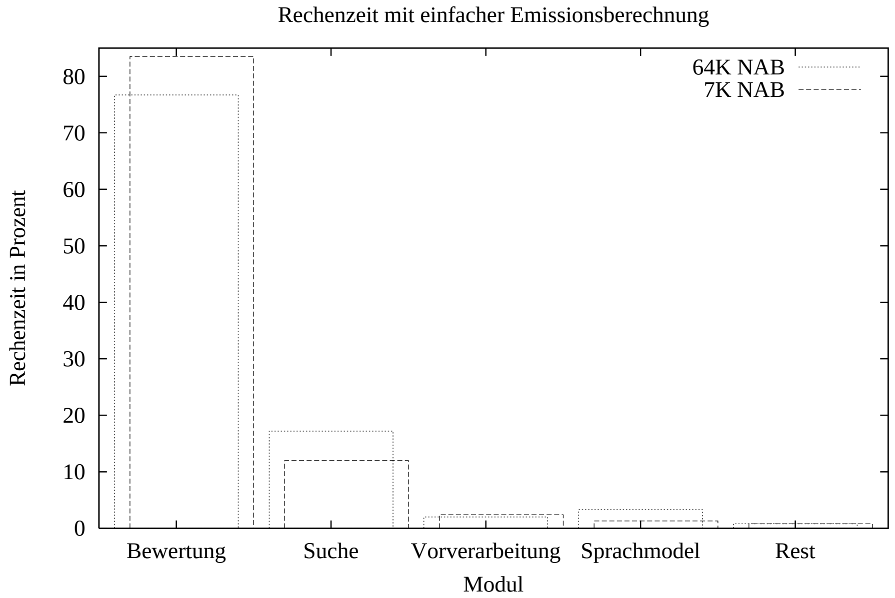

Abbildung 6: Rechenzeitverteilung bei einfacher Berechnung der Emissionswahrscheinlichkeiten für NAB

Echtzeit durch den Einsatz von Bucket-Box-Intersection. Die verschiedenen Punkte innerhalb einer Kurve in dieser Graphik stellen wieder verschiedene Einstellungen der Pruning-Schwellen des Systems dar. Da die Ausgangserkennungsleistung bei NAB besser ist als bei VM-Deutsch, können die BBI-Bäume agressiver berechnet

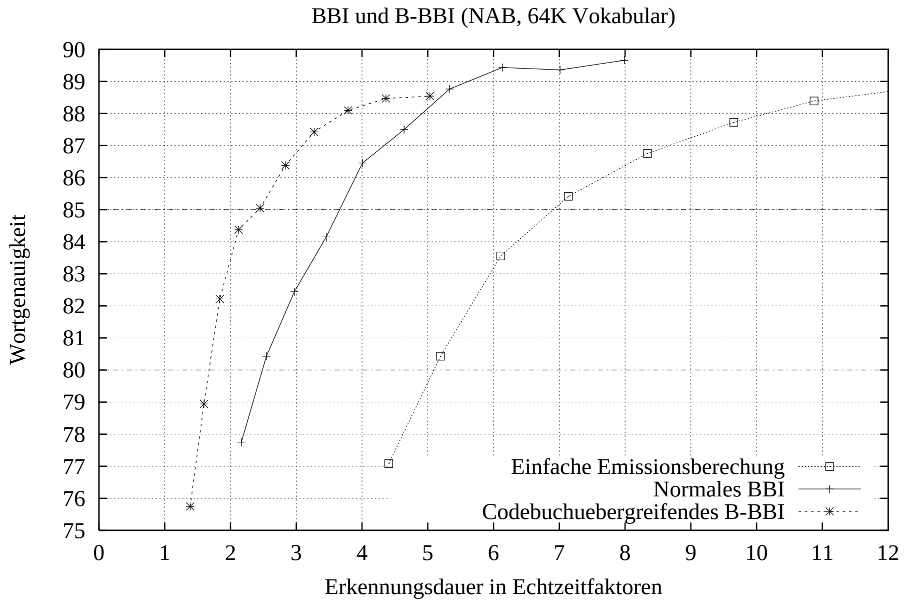

Abbildung 7: Bucket-Box-Intersection und codebuchübergreifende Big-Bucket-Box-Intersection für NAB mit 64000 Wörtern im Vokabular.

werden, bei größerem Verlust an Genauigkeit. Mittels der codebuchübergreifenden Variante kann eine Rechenzeit von weniger als 2.5 mal Echtzeit erreicht werden. Da die NAB-Codebücher deutlich größer als die VM-Deutsch-Codebücher sind, wurden hier größere Bäume der Tiefe 6 (BBI) und 10 (B-BBI) verwendet.

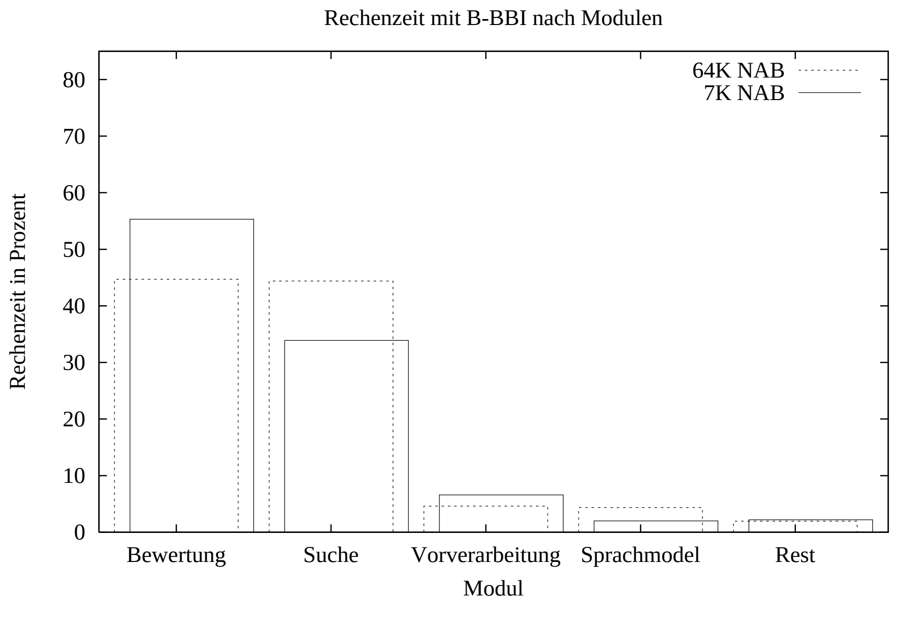

Abbildung 8: Rechenzeitverteilung mit B-BBI für NAB

In Abbildung 8 sieht man die Rechenzeitverteilung bei der Verwendung von B-BBI. Der Anteil der Berechnung der Emissionswahrscheinlichkeiten ist gegenüber dem System mit einfacher Berechnung der Emissionswahrscheinlichkeiten (Abbildung 6) von 78% auf 44% gefallen.

5 Zusammenfassung

Das codebuchübergreifende Bucket-Box-Intersection-Verfahren konnte erfolgreich von einem englischen Diktier-Szenario mit sehr großem Vokabular (NAB) auf ein Deutsches Terminabsprache-Szenario (VM-Deutsch) mit mittlerem bis großen Vokabular portiert werden. Der Gewinn an Erkennungsgeschwindigkeit beträgt im Vergleich zu der ursprünglich verwendeten Methode zur Berechnung der Emissionswahrscheinlichkeiten mehr als einen Faktor zwei. Für die kleineren Codebücher des Systems für die Deutsche Terminabsprache sind die Vorteile von B-BBI zu BBI noch deutlicher als für das englische NAB Diktiersystem. Das gleiche Verfahren wurde mit Erfolg auch für die Integration des Japanischen und Englischen Verbmobil-Erkenners verwendet.

Literatur

[EUR, 1995] *European Conference on Speech, Communication and Technology*, Madrid, Spain.

[ICA, 1996] *International Conference on Acoustics, Speech and Signal Processing*, Atlanta, USA. IEEE.

[Fritsch und Rogina, 1996] J. Fritsch und I. Rogina. The Bucket Box Intersection (BBI) Algorithm for fast approximative evaluation of Diagonal Mixture Gaussians. In *Proceedings of the ICASSP 1996 [ICA, 1996]*, Band 1, S. 837–840.

[Fritsch et al., 1995] J. Fritsch, I. Rogina, T. Sloboda und A. Waibel. Speeding up the Score Computation of HMM Speech Recognizers with the Bucket Voronoi Intersection Algorithm. In *Proceedings of the EUROSPEECH 1995 [EUR, 1995]*, Band 2, S. 1091–1094.

[Ramasubramanian und Paliwal, 1992] V. Ramasubramanian und K. Paliwal. Fast K–Dimensional Tree Algorithms for Neares Neighbor Search with Application to Vector Quantization Encoding. *IEEE Transactions on Signal Processing*, S. 518–531.