Ein System zur Interpretation einer Äußerung: Methoden

T. Kuhn, S. Kunzmann*, F. Kummert+, M. Mast, H. Niemann, E. Nöth, R. Prechtel, S. Rieck, A. Reißer, G. Sagerer+, E.G. Schukat-Talamazzini, J. Unglaub

Lehrstuhl für Informatik 5 (Mustererkennung), Universität Erlangen-Nürnberg
*IBM Deutschland GmbH, Wissenschaftliches Zentrum Heidelberg
+Arbeitsgruppe Angewandte Informatik, Universität Bielefeld

1 Einführung

Bei dem Spracherkennungs- und Dialogsystem EVAR [Nie91], dessen Akronym sich aus den Begriffen Erkennen, Verstehen, Antworten und Rückfragen ableitet, handelt es sich um ein Auskunftssystem für den Diskursbereich *"Intercity-Zugauskunft"*. Der Benutzer soll mit dem System in einem telefonischen Auskunftsdialog Informationen über Intercity-Verbindungen und Fahrpreise einholen können. Auf Grund der gegebenen Problemstellung ergeben sich die folgenden Randbedingungen: Beschränkung des Eingabesignals auf Telefonbandbreite, Sprecherunabhängigkeit, Erkennen und Verstehen von kontinuierlich gesprochenen Äußerungen sowie die Behandlung möglichst allgemeiner Auskunftsdialoge. Die Architektur des Systems orientiert sich an einem geschichteten linguistischen Modell. Das Zusammenwirken der Module läßt sich in die Verarbeitungsschritte Erkennen und Verstehen der Äußerung, sowie einer vom aktuellen sprachlichen und situationellen Kontext abhängigen Dialogkontrolle gliedern.

2 Die Erkennungsphase

Die Erkennungsphase in EVAR gliedert sich in die sequentiell ablaufenden Schritte Aufnehmen und digitale Vorverarbeitung des Sprachsignals, Merkmalberechnung, Lautsegmentierung und -klassifikation sowie Generierung von Worthypothesen. Die Schnittstelle zur linguistischen Analyse bildet die generierte Worthypothesenmenge.

Die Äußerungen werden an einem PC-AT mit 16 kHz abgetastet, mit 12 Bit quantisiert und auf den Bereich 0.1 - 6.4 kHz bandbegrenzt. Die weitere Analyse der Äußerungen findet auf einer RISC-Workstation unter dem Betriebssystem UNIX statt. Die Kommunikation zwischen PC und Workstation erfolgt unter Verwendung eines *Client-Server-Modells* auf der Basis des TCP-IP Protokolls, so daß der Datenaustausch in Echtzeit stattfinden kann [Kun91].

Für die Merkmalberechnung wird zunächst eine 512 Punkte FFT für alle 10 ms Fenster (Frame) berechnet. Die Fenster sind nicht überlappend. Die resultierenden 256 Werte des Leistungsspektrums werden über eine 64 Kanal Bark-Skala zusammengefaßt. Das Ergebnis wird logarithmiert und einer Cosinus-Transformation unterworfen, wobei die ersten 32 Koeffizienten zu 12 Werten weiter zusammengefaßt werden. Unter Berücksichtigung einer

Nachbarschaft von 5 Frames wird zu jedem der 12 Werte die Steigung der Regressionsgeraden berechnet. Sie dient als Maß für den zeitlichen Verlauf der cepstralen Merkmale. Das Resultat der Merkmalberechnung ist ein Vektor mit 24 Komponenten.

Die Frames werden im nachfolgenden Analyseschritt unter Verwendung eines Normalverteilungsklassifikators nach Lautkomponenten klassifiziert. Jede Lautkomponente ist durch eine multivariate Gauß'sche Verteilung mit voller Kovarianzmatrix beschrieben. Jedem Frame werden die 5 besten Lautkomponentenhypothesen zusammen mit ihren Bewertungen zugeordnet. Derzeit werden 49 Lautkomponentenklassen unterschieden.

Bei der Lautsegmentierung und -klassifikation wird in der ersten Alternative des Lautkomponentenstromes nach homogenen Bereichen gesucht. Bedingt durch koartikulatorische Einflüsse kann davon ausgegangen werden, daß die Elemente dieser initialen Segmentierung deutlich kürzer als Laute sind. Durch Betrachten potentieller Segmentanfänge und -enden wird ein Segmentgraph erzeugt, dessen Elemente mit *Hidden-Markov-Modellen* (HMM) in lautliche Einheiten klassifiziert werden. Mit einem heuristischen Graphsuchverfahren (A^*-Algorithmus) wird die optimale Lautfolge gefunden. Unterschieden werden zur Zeit 45 Laute, die durch 51 kontextunabhängige HMM's modelliert werden.

Die Generierung der Worthypothesen erfolgt unter Verwendung von diskreten HMM's. In der Worthypothese sind die Informationen bestehend aus dem Wort selbst, die bzgl. des Sprachsignals hypothetisierte Anfangs- und Endeframeadresse, sowie eine als Gütekriterium anzusehende Bewertung zusammengefaßt. Um die Modellparameter adäquat trainieren zu können, wird für jeden Lexikoneintrag das Wortmodell aus der phonetischen Umschrift nach Lauten erzeugt. Jeder auftretende Laut wird durch ein kantenorientiertes HMM mit 2 Zuständen und 3 Kanten modelliert. Diese elementaren HMM's ermöglichen die Modellierung der Ersetzung, Einfügung und Auslassung einer Beobachtung. Das Wortmodell ergibt sich durch Konkatenation der elementaren HMM's. Das Verfahren zur Worthypothesengenerierung basiert auf dem *One-Stage-Algorithmus* [Ney84]. Das Lexikon wurde daher als Baum repräsentiert, wobei jedes Wortende mit dem Wurzelknoten des Lexikonbaumes verbunden ist. Zu jedem Zeitpunkt (hier: Segment) werden die 30 besten Wortketten bestimmt und das letzte Wort der gefundenen Kette als Worthypothese generiert. Als Bewertung der Worthypothese wird die bzgl. der Anzahl bislang berechneter Zeitscheiben normierte α-Wahrscheinlichkeit verwendet.

3 Linguistische Analyse

Das für die Interpretation einer Äußerung benötigte linguistische und anwendungsabhängige Wissen ist mit Hilfe der Wissensrepräsentationssprache ERNEST (ERlanger semantisches NEtzwerk SysTem) [Nie90] als semantisches Netz dargestellt. Die homogene Wissensbasis ist in verschiedene Abstraktionsebenen gegliedert (siehe Abb. 1). Die unterste Ebene stellt die Schnittstelle der linguistischen Analyse zur Worthypothesengenerierung dar. In der Syntaxebene werden einzelne Wortarten und Satzkonstituenten repräsentiert. Ebenso ist in den Konzepten dieser Ebene Information über die syntaktische Konsistenz repräsentiert. Das semantische Wissen ist auf Grundlage von Fillmore's Tiefenkasustheorie modelliert. Für die im Anwendungsbereich benötigten Verben und Substantive wurden Kasusrahmen modelliert, die angeben, wie die Leerstellen, die das Verb oder Substantiv eröffnet, mit Tiefenkasus zu besetzen sind. Jeder Kasusrahmen sowie jeder Tiefenkasus ist

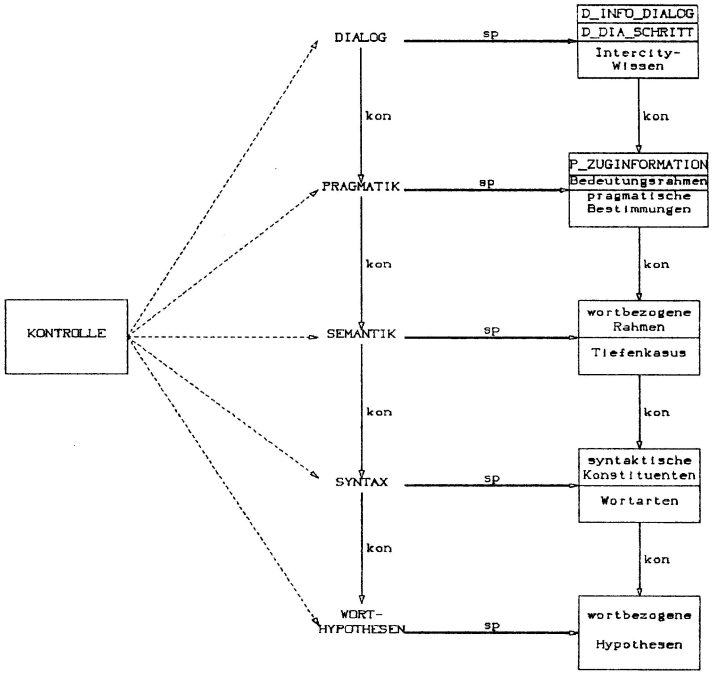

Abbildung 1: Realisierung der homogenen Wissensbasis in ERNEST

durch ein Konzept repräsentiert. In der Pragmatikebene ist das anwendungsspezifische Wissen für den Bereich *'Intercity-Auskunft'* modelliert, wobei jeder mögliche Auskunftstyp sowie die dadurch vorgegebenen pragmatischen Bestimmungen durch Konzepte dargestellt sind. In der Dialogebene wird ein Auskunftsdialog durch ein Dialogmodell bestehend aus Dialogschritt-Typen dargestellt. Die einzelnen Abstraktionsebenen sind über Konkretisierungskanten miteinander verbunden. Da in ERNEST eine problemunabhängige prozedurale Semantik definiert ist, wird neben der Modellierung einer Wissensbasis auch ein geeigneter Kontrollalgorithmus zur Verfügung gestellt. Die Kontrolle operiert auf dem Netzwerk. Dabei werden Konzepte datenabhängig modifiziert und Instanzen gebildet. Ziel der Analyse ist die Instanziierung eines Dialogschritt-Typs, der einerseits dem zu Grunde gelegten Sprachmodell gerecht wird und andererseits eine maximale Verträglichkeit mit der akustischen Eingabe zeigt. Dazu wird ausgehend von der Worthypothesenmenge, eine *bottom-up* und *top-down* gemischte Analysestrategie unter Einbeziehung des linguistischen Wissens der Wissensbasis angewandt.

4 Charakterisierung der Vorführversion

Zum Testen der einzelnen Analyseschritte und zum Integrieren neuer Methoden wurde eine Arbeitsumgebung realisiert, welche eine vollständige Verarbeitung vom Sprachsignal

bis zur Antwortgenerierung erlaubt. Die Dialogsteuerung beschränkt sich zur Zeit auf die Zuordnung der pragmatischen Interpretation der Äußerung zu einem Dialogschritt-Typ. Der Dialog in der Arbeitsumgebung besteht momentan aus der initialen Informationsabfrage des Benutzers und der Antwort des Systems. Handelt es sich bei der Äußerung um eine Anfrage nach einer Intercity-Verbindung, kann eine passende Verbindungsauskunft generiert und über ein Sprachsynthesegerät ausgegeben werden.

Für Demonstrationszwecke wurde eine Vorführversion in der Arbeitsumgebung realisiert, wobei zur Vereinfachung des Erkennungsprozesses die unterschiedlichen Parameter sprecherabhängig trainiert wurden. Für die Vorführversion wurden die Parameter anhand von 200 phonetisch balancierten Sätzen (ca. 9 Minuten) und 100 (ca. 7 Minuten) anwendungsabhängigen Äußerungen trainiert. Als Wortschatz wurde ein Lexikon mit 1073 Einträgen definiert. Auf einem RISC-Rechner (DECStation 5200, 25 mips) benötigte die Analyse des Dialogs

Anfrage:	*"Wann kann ich morgen früh nach München fahren?"*
Systemantwort:	*"Sie können um neun Uhr neunundvierzig in Nürnberg abfahren."*
	"In München sind Sie dann um elf Uhr zweiunddreißig."

von der Anfrage bis zur automatisch generierten Systemantwort 76.9 Sekunden CPU-Zeit (siehe Tabelle 1). Dies bedeutet einen ungefähren Echtzeitfaktor von 21.

Analyseschritte	Verweilzeit in Sekunden	CPU-Zeit in Sekunden
Merkmalberechnung	5.3	4.9
Lautkomponentenklassifikation	5.9	4.9
Lautklassifikation	1.8	1.1
Worthypothesengenerierung	8.3	6.2
Linguistische Analyse	105.9	59.8
	127.2	76.9

Tabelle 1: Verweilzeit und CPU-Zeit für eine Äußerung

Literatur

[Kun91] S. Kunzmann, T. Kuhn, E. Nöth, G. Stallwitz: *Ein System zur Interpretation einer Äußerung: Realisierung.* In *Fortschritte der Akustik-DAGA '91*, 1991.

[Ney84] H. Ney: *The Use of a One-stage Dynamic Programming Algorithm for Connected Word Recognition.* IEEE Trans. on Acoustics, Speech and Signal Processing, ASSP-32: S. 263–271, 1984.

[Nie90] H. Niemann, G. Sagerer, S. Schröder, F. Kummert: *ERNEST: A Semantic Network System for Pattern Understanding.* IEEE Trans. on Pattern Analysis and Machine Intelligence, S. 883–905, 1990.

[Nie91] H. Niemann: *The combination of word recognition and linguistic processing in speech understanding.* In *NATO ASI Speech Recognition and Understanding: Recent Advances, Trends and Applications*, Springer-Verlag, Berlin, erscheint 1991.