Methoden

WIE SAGE ICH ES MEINEM COMPUTER?

Foto: Fraunhofer

der automatischen Spracherkennung

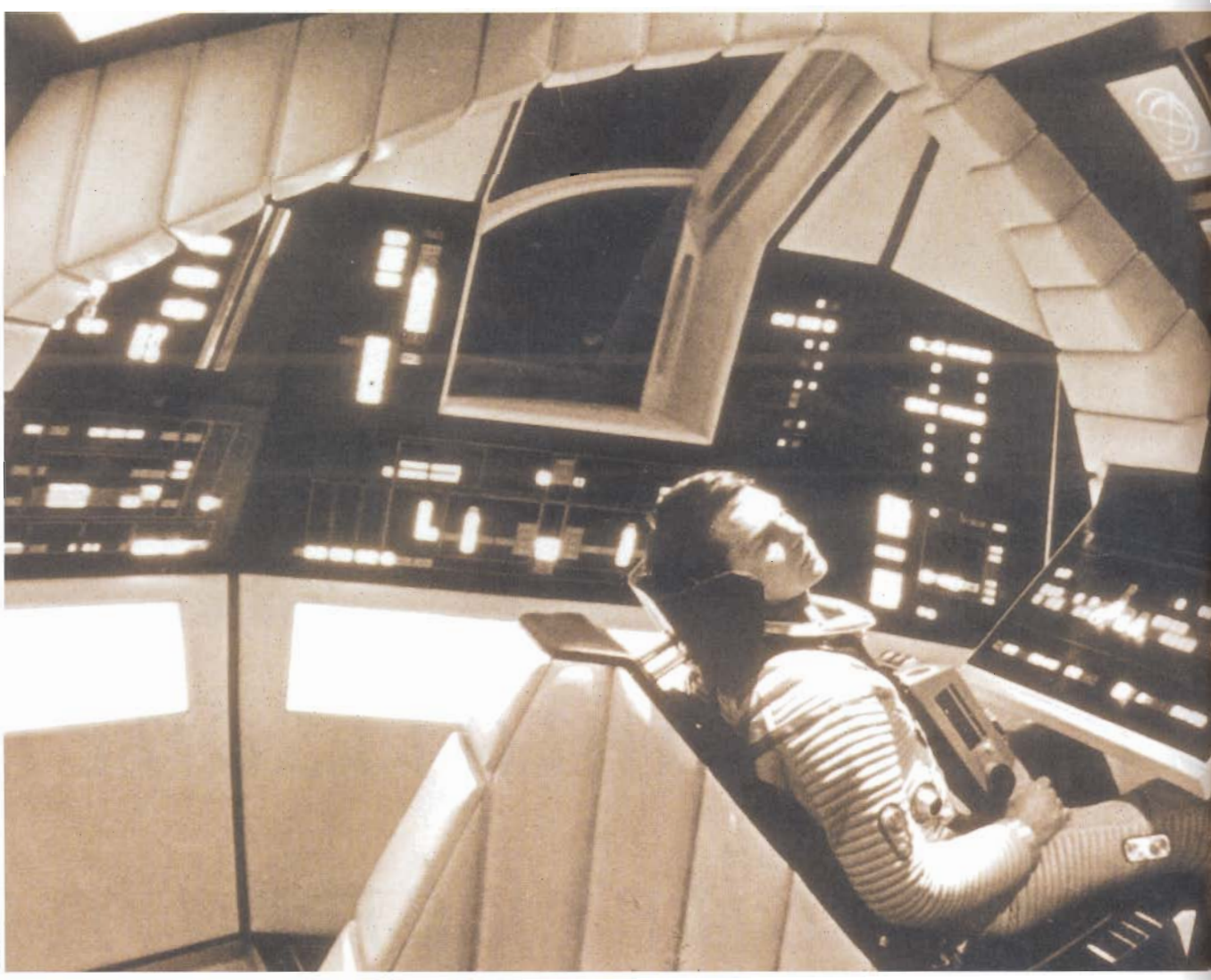
Raúl Rojas

In „2001 Odyssee im Weltraum", einem der besten Science-Fiction Filme aller Zeiten, spielt ein Computer eine der Hauptrollen. Der Rechner „HAL" kann sich mit dem Raumfahrer unterhalten, Schach spielen, von seinen Lippen ablesen und reagiert sogar emotional. Die Crew spricht mit „HAL" wie mit einem Kollegen. Wenn aber der mörderische Computer abgeschaltet wird, geht sein Sprachvermögen langsam verloren, bis er in einer dramatischen Szene schließlich verstummt.

Einen Computer wie „HAL" zu bauen, einen Computer, der Sprache versteht, ist ein alter Traum der Wissenschaftler. Dabei geht es nicht darum, einen Computer zu entwickeln, der alle Nuancen der Sprache verstehen und deuten kann – es geht um etwas viel Einfacheres: Um die Diktiermaschine der Zukunft, einen Computer, der unsere Briefe aufnehmen und transkribieren kann. Systeme für die automatische Spracherkennung bringen den Computer näher zu den Menschen, weil der Benutzer nicht mehr über eine Tastatur mit dem Computer kommuniziert, sondern in seinem eigenen Code, das heißt mit Worten. Spracherkenner sind aber noch weit davon entfernt, wie „HAL" zu arbeiten. Fast perfekte automatische Spracherkennung bleibt noch ein offenes Problem und ist ein sehr aktives Forschungsfeld.

Dabei unterscheidet man zwei Arten der Spracherkennung: Sprecherabhängig und Sprecherunabhängig. Im ersten Fall wird das System auf die Stimme und Diktion einer Person trainiert, und andere Personen können das System erst nach einer Anpassung verwenden. Das Ziel der heutigen Forschung ist, Systeme zu bauen, die nicht maßgeschneidert für eine Person sind, das heißt vom Sprecher unabhängig. Dabei soll das System auch mit fließend gesprochenen Sätzen umgehen können. Der Benutzer soll nicht jedes Wort mit ... einer ... künstlichen ... Pause ... voneinander ... trennen, sondern soll natürlich reden können.

Der Rechner „Hal"

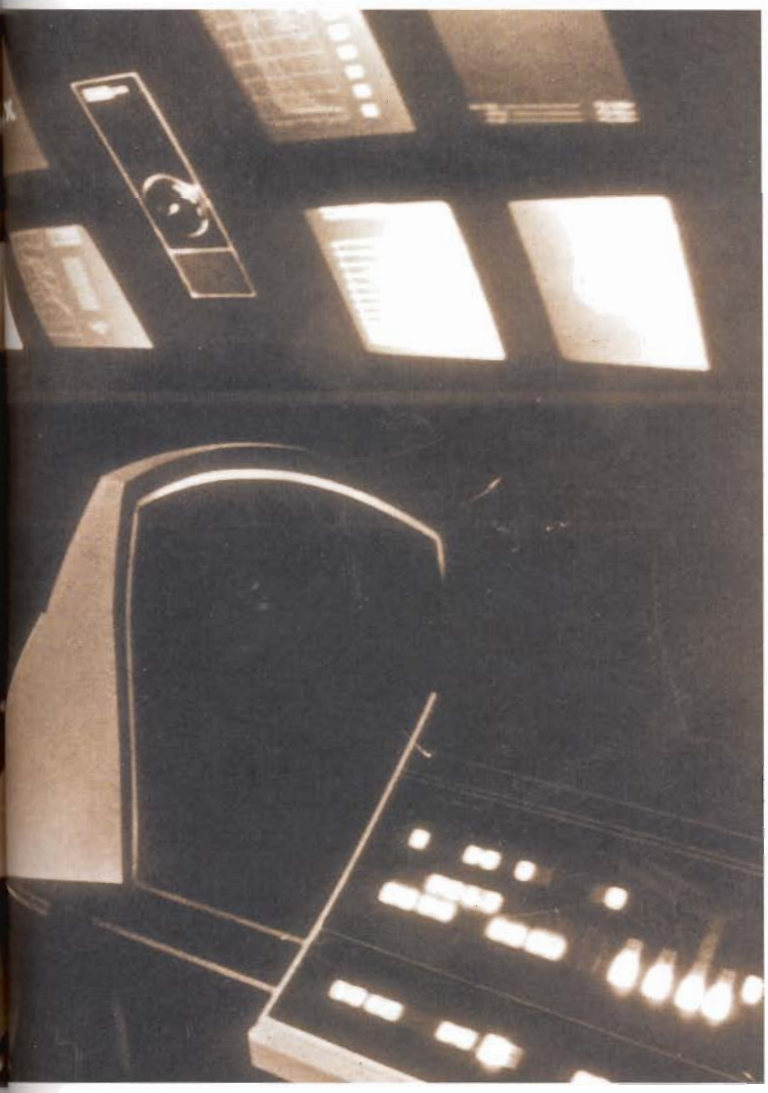


Ullstein Bilderdienst

In diesem Beitrag möchte ich erläutern, warum automatische Spracherkennung immer noch ein schwieriges Problem ist und welche Methoden in der Informatik verwendet werden, um Computern das Verstehen der Sprache beizubringen. Automatische Spracherkennung ist ein wahrlich interdisziplinäres Feld, wo Methoden aus der Linguistik, der Signalverarbeitung und der Mustererkennung zum Einsatz kommen.

Das akustische Signal

Bei der automatischen Spracherkennung geht es darum, ein akustisches Signal in den entsprechenden Text zu transformieren. Der erste Schritt besteht also darin, die Sprache mit einem Mikrophon aufzunehmen. Wir wissen aus physiologischen Untersuchungen, dass Menschen akustische Signale in ihre elementaren Bestandteile zerlegen, das heißt die Frequenzmischung wird im Ohr und Gehirn analysiert. Eine ähnliche Art der Informationsverarbeitung kann man mit einem Spektrogramm durchführen: Dabei wird Sprache sicht-



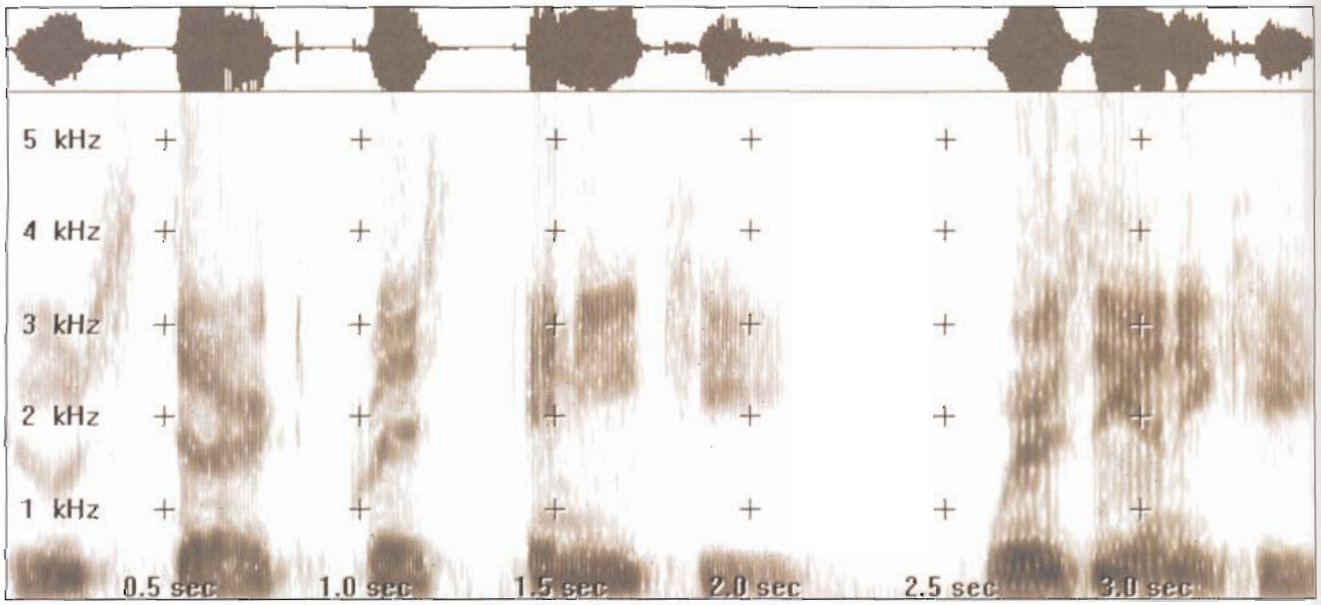


Abbildung 1: Spektrogramm aus den Mikrofondaten

bar für das Auge gemacht (womit wir mehr über den Prozess der Spracherzeugung erfahren können) und ähnliche Daten werden dann dem Computer übergeben.

Ein Spektrogramm ist eine Zerlegung des akustischen Signals in seine Frequenzkomponenten. Abbildung 1 zeigt ein Beispiel: In die waagerechte Richtung verläuft die Zeit. In die senkrechte Richtung wird mit Grauwerten die Intensität von jeder vorhandenen Frequenz angezeigt. Die gemusterten Frequenzen gehen von 0 bis 5 KHz. Dort, wo eine Frequenz besonders stark hervortritt, wird geschwärzt. Ist eine Frequenz nicht vorhanden, bleibt das entsprechende Feld hell. Ganz oben im Bild sieht man die Mikrophondaten: Schwingungen entsprechen den in drei Sekunden ausgesprochenen Worten. Darunter sieht man das Spektrogramm in der Zeit. Die senkrechten Streifen stammen aus dem Musterungsprozess (die Frequenzen werden alle 50 Mikrosekunden ermittelt). Man sieht sehr deutlich, dass sich gewisse waagerechte Muster bilden, dunkle Streifen, die sich nach oben (in die höheren Frequenzen) oder nach unten bewegen. Diese dunklen Streifen werden „Formanten" genannt und ihre Position und Anzahl ist bereits ein wichtiges Indiz für die Art des Lautes, der ausgesprochen wurde. Geübte Linguisten können sogar manchmal aus dem Spektrogramm auf die gesprochenen Worte schließen.

Insbesondere Vokale zeigen eine deutliche Bänderstruktur in dem Spektrogramm. Abbildung 2 zeigt ein Beispiel des Spektrogramms, wenn „e-a-e-a" ausgesprochen wird. Man sieht die Übergänge von einem Vokal in den anderen, und man sieht deutlich, dass die Formanten sich auf unterschiedlichen Höhen befinden.

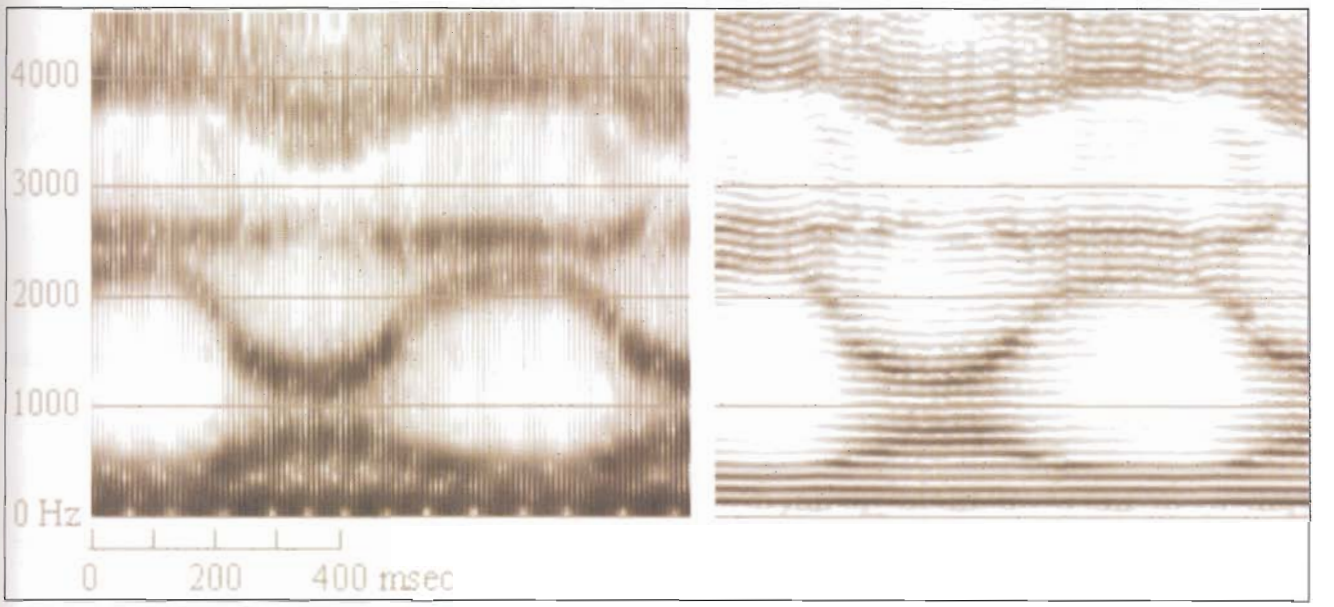


Abbildung 2: Formanten der Vokale: e – a – e – a

Das Phänomen des Übergangs der Formanten eines Lautes in den nächsten macht automatische Spracherkennung schwierig, da es nicht immer klar ist, wo ein Laut aufhört und wo der nächste beginnt. Andere Laute, insbesondere solche wie „sch" und „f", erzeugen Spektrogramme, die eher wie Rauschen aussehen. Bei Konsonanten ist die Formantenstruktur nicht besonders ausgeprägt.

Bei der automatischen Spracherkennung sind viele unterschiedliche Ansätze getestet und implementiert worden. Eine populäre Methode besteht darin, die akustischen Signale für bestimmte Worte zu speichern und sie als Schablone zu benutzen. Dabei werden das Spektrogramm oder Varianten davon verwendet (wie zum Beispiel das cepstrum, mel cepstrum, usw.). In der Zeitachse werden zum Beispiel alle 100 ms zehn der Frequenzbänder gemustert und die Intensitätswerte gespeichert. So verwandelt sich ein Wort, dass in einer halben Sekunde ausgesprochen wurde, in einen Datensatz mit 500 Einträgen. Wird jetzt ein neues Wort ausgesprochen, kann das Spektrogramm wieder berechnet werden und die Ähnlichkeit beider Worte kann getestet werden. Wir speichern also Schablonen für die Worte im Wörterbuch („ja" oder „nein" für ein einfaches Auskunftssystem) und vergleichen neue Worte mit diesen Schablonen. Es wird das Wort ausgewählt, das den kleinsten Abstand zu dem Signal hat. Diese Methode hat jedoch den Nachteil, dass nur eine begrenzte

Erste Erkennungsmethode: Schablonen

Prof. Dr. Raúl Rojas

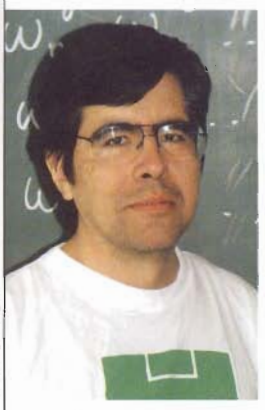


Foto: Aufschlager

Raúl Rojas, geboren in Mexiko Stadt, studierte Mathematik und Physik an der Nationalen Technischen Universität Mexikos. Abschluss des Mathematikstudiums mit dem Master in Science. Parallel dazu Master in Economics an der Autonomen Nationalen Universität Mexikos. Promotion und Habilitation an der FU Berlin. Von 1994 bis 1997 C3 - Professor für künstliche Intelligenz an der Universität Halle. Seit Ende 1997 C4 - Professor an der Freien Universität Berlin. Rojas ist Herausgeber mehrerer Bücher, u.a. „Neural Networks", Springer-Verlag, 1996; „The First Computers", MIT-Press, Cambridge 2000; „Encyclopedia of Computers and Computer History", Fitzroy-Deaborn, NY, 2001.

Kontakt:
Fachbereich Mathematik und Informatik
Institut für Informatik
Takustr. 9
14195 Berlin
Tel. 030 / 83 87 51 30
E-Mail: rojas@inf.fu-berlin.de

Anzahl von Schablonen verarbeitet werden kann. Darüber hinaus ist die Methode sprecherabhängig und kann nur isolierte Wörter aus einem Wörterbuch erkennen, jedoch keine fließend gesprochenen Texte.

Ein Problem ist außerdem zu lösen: dasselbe Wort kann unterschiedlich schnell ausgesprochen werden, einmal in 0,5 Sekunden, ein anderes mal in 0,4 Sekunden. Wir möchten das Wort nicht zurückweisen, nur weil es diesmal schnell ausgesprochen wurde. Der Vergleich der Schablonen wird deswegen mit Hilfe der sogenannten „time warping"-Methode durchgeführt. Das bedeutet, dass der Computer die gespeicherten Schablonen optimal streckt oder staucht (an den passenden Stellen), um die maximal mögliche Übereinstimmung zu ermitteln. Die lange Schablone (0,5 Sekunden Sprachdaten) wird in diesem Beispiel so gestaucht, dass sie am besten zu den 0,4 Sekunden Sprachdaten passt. Die Schablone, die am besten passt, wird danach ausgewählt. Nachteil des Verfahrens ist jedoch, dass auch bei unsinnigen Vergleichen (das Wort „bitte" wird zum Beispiel mit „ja" oder „nein" verglichen) das System sich für eine der vorhandenen Alternativen entscheidet. Daraus können grobe Fehler in dieser Art von Dialogsystemen entstehen, und ein Bahninformationssystem möchte uns nach Saarbrücken statt nach Unterhaching schicken.

Will man von dem einfachen Vergleich gespeicherter Schablonen wegkommen, der sprecherabhängig ist und nur isolierte Worte behandelt, so bietet sich die Verwendung eines Klassifikators an, der ständig versucht, Laute anhand des Spektrogramms zu klassifizieren. Wir möchten also eine Art Mustererkennung über das Sprachsignal durchführen. Hierfür ist es nützlich, eine Vorverarbeitung des Signals durchzuführen. Eine häufig verwendete Methode besteht darin, das Spektrogramm in kleine Segmente (Fenster) zu teilen, die zum Beispiel nur einige Mikrosekunden lang sind. Sind dann in einem Fenster die Frequenzen F1, F2, ..., F18 mit Intensitäten A1, A2, ..., A18 vorhanden, dann möchten wir wissen, wie hoch die Wahrscheinlichkeit ist, dass einer von 64 Phonemen (Lauten) ausgesprochen wurde. Dies kann mit einem Klassifizierungsnetz geleistet werden. Die Phoneme sind von Linguisten definiert worden und decken alle sprachlichen Kombinationen ab.

Zweite Variante: Mustererkennung

Abbildung 3 zeigt die Architektur eines Netzes, das bei einem an der Universität von Kalifornien entwickelten Spracherkenner zum Einsatz gekommen ist [Bourlard, Morgan 1993]. Die Ausgabe des Netzes besteht aus 64 Ausgabewerten, die anzeigen sollen, welches von 64 Phonemen in das Netz eingegeben wurde. Idealerweise sollte nur eine der Ausgabeleitungen auf Eins schalten, während die anderen „stumm'" (das heißt bei Null) bleiben. Die Sprachsignale werden segmentiert und von jedem Fenster werden 18 Koeffizienten genommen. Dies können die Amplituden im Spektrogramm sein, wobei nur 18 unterschiedliche Frequenzbereiche berücksichtigt werden. Es können aber auch zusätzliche Transformationen der Sprachsignale durchgeführt werden, die aber für unsere Zwecke unerheblich sind. Jedoch werden in das Netz nicht nur die 18 Koeffizienten eines Fensters eingegeben, sondern auch der „Kontext" des Signals. Dieser besteht aus den Koeffizienten der sechs Fenster vor und der sechs Fenster

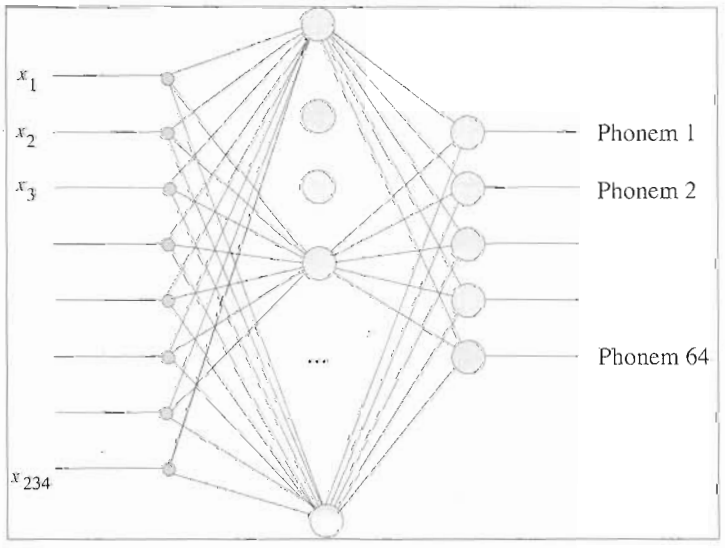


Abbildung 3: Bayes-Netz für die Klassifikation der Phoneme

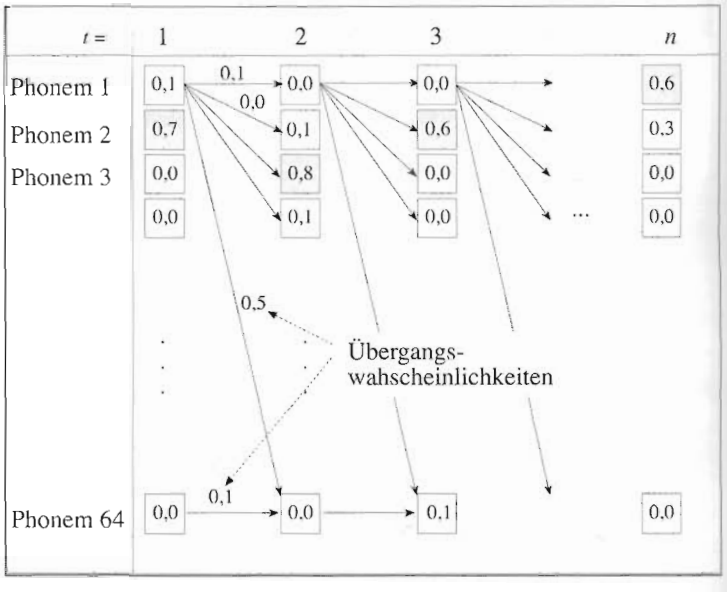


Abbildung 4: Bestimmung des Pfades maximaler Wahrscheinlichkeit

nach dem aktuellen Fenster. Dies ergibt insgesamt 13 mal 18, also 234 Eingabewerte für das Netz.

Das gezeigte Netz wird trainiert, und zwar werden bekannte Sprachsignale segmentiert und das Netz präsentiert. Der „Trainer" weiß in jedem Moment, welches Phonem ausgesprochen wurde und verändert die Parameter des Klassifikators (im wesentlichen die Stärke der Verbindungskanten), um das Netz zu zwingen, eine „1" bei der Ausgabeleitung – die zum ausgesprochenen Phonem gehört – und „0" bei allen anderen Leitungen zu erzeugen. Dies wird mit Hunderten oder Tausenden von Phonemen wiederholt, bis das Netz gelernt hat, diese Daten zu klassifizieren. Dafür existieren die entsprechenden Algorithmen.

Interessant ist jetzt, dass sich das Netz bei der Präsentation eines unbekannten Signals für die wahrscheinlichste Alternative entscheidet. Kommt zum Beispiel ein „a" über das Mikrophon, wird die Ausgabeleitung für diesen Laut am höchsten steigen, und es wird auf diese Weise angezeigt, dass wahrscheinlich ein „a" vorliegt. Der Klassifikator steigt aber selten bis zu „1" bei der Ausgabe, er liefert nur Werte zwischen 0 und 1 (z. B. 0,7). Diese Werte können als die Wahrscheinlichkeit des Auftretens des Lautes „a" interpretiert werden. Andere Laute bekommen die Restwahrscheinlichkeit zugewiesen.

Die Art von Klassifikator ist ein Beispiel eines neuronalen Netzes. Für das Training des Netzes werden viele Stunden sprachlicher Daten verwendet (häufig werden diese Daten in CDs gepresst und verteilt). Da man für diese Daten die korrekte Transkription kennt, ist es relativ einfach, die Korrespondenz zwischen Text und segmentierten Daten herzustellen. Dies kann von einem Linguisten manuell gemacht werden. Wegen der hohen Anzahl der Parameter des Klassifikators und der großen Trainingsmenge nimmt der Lernvorgang viel Zeit in Anspruch. Durch schnelle parallele Hardware können aber die Lernzeiten auf ein vernünftiges Maß reduziert werden.

Im normalen Betrieb kann das Klassifizierungsnetz verwendet werden, um die Reihenfolge der Wahrscheinlichkeit der Phoneme zu ermitteln. Das heißt, bei einem Mikrophonsignal kann der Klassifikator für jeden Zeitpunkt bestimmen, wie groß die Wahrscheinlichkeit ist, dass beim Zeitpunkt 1 Phonem „a" oder „e" oder „t" usw. ausgesprochen wurden. Dasselbe gilt für den Zeitpunkt 2, 3, usw. Der Spracherkenner muss dann die optimale Reihenfolge auswählen. Wir wissen zum Beispiel, dass ein „a" lang ausgesprochen werden kann. Das bedeutet: Mit großer Wahrscheinlichkeit kann auf ein „a" zum Zeitpunkt 1 auch ein „a" zum Zeitpunkt 2 folgen. Dagegen ist es schwierig, dass auf ein „t" zum Zeitpunkt 1 ein „k" zum Zeitpunkt 2 folgt. Der Spracherkenner baut also auf statistisches Wissen über die zu erkennende Sprache auf. Linguisten liefern diese Daten und bilden Wortmodelle, womit die optimale Auswahl der Phoneme gemacht werden kann.

Bestimmung des Pfades maximaler Wahrscheinlichkeit

Abbildung 4 zeigt, wie die optimale Lautsequenz ausgewählt wird. Zu jedem Zeitpunkt gibt es eine Mischung

von Wahrscheinlichkeiten für jeden Laut. Die Pfeile zeigen an, wie groß die Wahrscheinlichkeit eines Übergangs von einem Laut zu einem anderen ist. Für jeden Zeitpunkt haben wir 64 bedingte Phonem-Wahrscheinlichkeiten. Alle diese 64 Werte werden mit der nächsten Spalte von Phonem-Wahrscheinlichkeiten verbunden, wobei die Wahrscheinlichkeit eines Übergangs als Gewicht jeder Kante notiert wird (nicht alle Verbindungen wurden gezeichnet, um die Abbildung übersichtlich zu halten). Durch die Methoden der dynamischen Programmierung oder anderer passender Algorithmen kann dann der Pfad der maximalen Wahrscheinlichkeit bestimmt werden (schattierte Felder). Dafür ist es wichtig, die Pfade, die verfolgt werden, passend zu segmentieren. Dies kann zum Beispiel durch „silence spotting" oder „word spotting" gemacht werden. Wo Pausen erkannt werden, kann der aktuelle Pfad abgebrochen und ein neuer Pfad gestartet werden. Dies verhindert, dass die Pfadwahrscheinlichkeit letztendlich zu stark abnimmt, so dass Alternativen sich nur gering voneinander unterscheiden. Selbstverständlich kann viel mehr Information in die Berechnung eingebracht werden. Mit Hilfe eines Wörterbuches können unsinnige Phonemkombinationen ausgeschlossen werden. Wird noch dazu die Grammatik der Sätze überprüft, können ungrammatikalische Sätze ebenfalls ausgeschlossen werden.

In dieser kurzen Darstellung haben wir lediglich das Thema angerissen. Es sollte aber klar sein, dass heutige Spracherkennungssysteme weit davon entfernt sind, den Sinn der ausgesprochenen Sätze zu verstehen. Es wird mit „brute force" gerechnet, ähnlich wie Computer heute Schach spielen: Statt wie der Weltmeister auf einmal ein Muster zu erkennen, werden alle Möglichkeiten durchprobiert und die beste wird ausgewählt. Das reicht schon heute, um den Schach-Weltmeister zu schlagen – es reicht aber noch nicht, um Sprache fehlerfrei zu erkennen. An der Freien Universität arbeiten wir an diesem Problem und zwar mit einem Ansatz, der vielversprechend erscheint. Wir bauen ein geschlossenes System, in dem ein Computer gleichzeitig lernt, einen Text vorzulesen und das Vorgelesene zu erkennen und in Text zu verwandeln. Dabei versucht der Computer nah an einem menschlichen Vorleser zu bleiben. Damit kann der Computer sich selbst trainieren und die aufwändige linguistische Vorbereitung der Daten entfällt. Der Computer unterhält sich mit sich selbst.

Automatische Spracherkennung hat sich von den Universitäten in die Softwarehäuser bewegt. Es gibt bereits verschiedene Systeme, die für Diktate oder für Auskunftssysteme verwendet werden können. Allerdings kann ein erfahrener Tipper schneller mit der Schreibmaschine als mit diesen Diktiermaschinen umgehen. Der Grund liegt in der noch großen Fehlerrate der Systeme: Korrekturen erfordern viel Zeit und stoppen den Fluss der Gedanken. Auch minimale Änderungen in der Umgebung (ein neues Mikrofon, Musik im Hintergrund, usw.) können die besten Spracherkennungs-

Garry Kasparow unterliegt IBM's Parallelrechner Deep Blue.

Foto: Ullstein

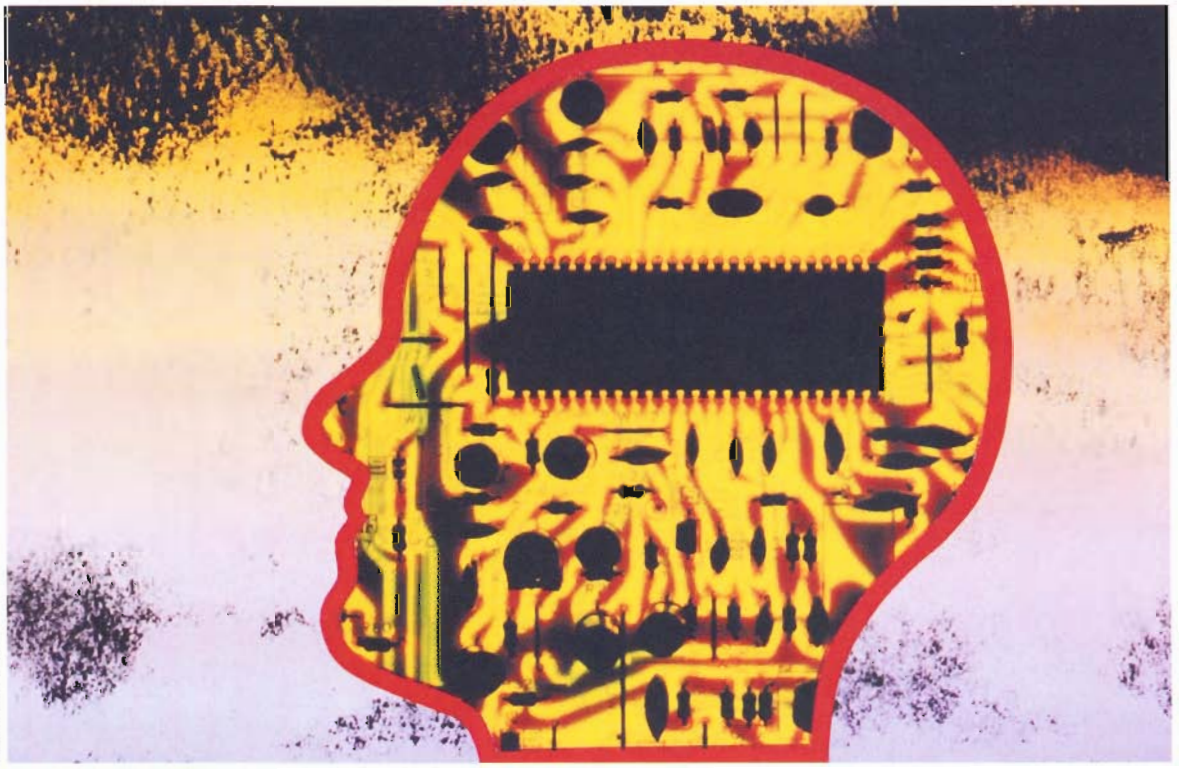


Illustration: Unicom

systeme in die Knie zwingen. Aber auch so wird erwartet, dass bis 2005 der Markt für Spracherkennungs-Software stetig zunehmen wird, und es wird bis 2005 mit Umsätzen in Milliardenhöhe gerechnet.

Das größte Problem für Spracherkennungssysteme ist die soziale Akzeptanz. Wenn ich mit meinem Computer kommuniziere, will ich wirklich mit dem Computer sprechen? Ist es nicht einfacher, Befehle mit der Maus oder mit der Tastatur einzugeben? Stören sprachliche Befehle am Computer meine Mitarbeiter nicht? Wahrscheinlich ist es so, dass für Computer am Arbeitsplatz die Eingabe von Befehlen über ein „künstliches" Medium wie Tastatur und Maus ergonomischer ist. Anders verhält es sich mit den kleineren tragbaren Computern, den Personal Assistants. Sie sind so klein, dass keine Tastatur mehr angeschlossen werden kann. Die Eingabe über Stifte ist auch nicht besonders schnell. Da diese Geräte in der Zukunft auch als Handy verwendet werden, bietet es sich an, sie gleich mit Spracherkennung auszustatten. Es wird deswegen erwartet, dass Spracherkennung vor allem für diese kleineren Geräte eingesetzt wird und dass PDAs die treibende Kraft für der Entwicklung neuer Software werden. Spracherkennungssoftware für diese Geräte muss jedoch um ein Vielfaches robuster sein als Software für Benutzung in einem Büro, da die Umgebung viel lauter ist.

Die Zukunft der automatischen Spracherkennung

Blinde und behinderte Benutzer können sicherlich bereits jetzt von Spracherkennungssoftware profitieren. Es ist zu erwarten, dass für diese Benutzer immer bessere Systeme erstellt werden.

Jedoch werden wir wahrscheinlich in den nächsten 20 Jahren immer noch nicht so wie in den Science-Fiction-Filmen mit den Computern sprechen können. Auch wenn alle Sprachverarbeitungsschritte beherrscht werden und Worte mit 99,9 % Genauigkeit erkannt werden sollten, wird es noch große Defizite im Sprachverständnis geben. Sprache ist sehr abhängig vom Kontext und bereits kleine Nuancen können den Sinn eines Satzes umdrehen. Ob etwas ironisch oder im Ernst gemeint ist, wird der Computer noch nicht erkennen können. Was allerdings nach diesen 20 Jahren möglich wird, ist schwer zu sagen. Man sollte sich hüten, in der Informationstechnik Prognosen zu liefern, die länger als 10 Jahre gelten. Die Grenze habe ich hier zweifach überschritten.

Literaturverzeichnis

Bourlard, H., Morgan, N. (1993), Connectionist Speech Recognition, Kluwer.

Rabiner, L., Bing-Hwang, J. (1993), Fundamentals of Speech Recognition, Prentice-Hall International, London.

Rojas, R. (1996), Neural Networks, Springer-Verlag, Berlin, New York.