Técnicas para a implementação de sistemas de reconhecimento automático de voz

José Luis Gómez Cipriano[1, 2], Roger Pizzatto Nunes[2], Dante Augusto Couto Barone[2]

[1]Instituto de Ciências Exatas e Tecnológicas - Centro Universitário Feevale; [2]Instituto de Informática - Universidade Federal do Rio Grande do Sul.

Resumo

O reconhecimento de voz é uma habilidade natural do ser humano, mas é uma tarefa difícil para os computadores e estações de trabalho atuais. Grandes avanços têm sido obtidos durante os passados 40 anos de história do reconhecimento automático de voz (RAV). Neste trabalho, descrevem-se as etapas do reconhecimento automático de voz e as técnicas que a maioria dos sistemas de RAV utilizam em cada etapa.

Palavras-chave

Reconhecimento de voz, reconhecimento de padrões, processamento digital de sinais.

Abstract

Speech recognition is a human natural ability, however it is a difficult task for current computers and work stations. Great advances have been obtained during the past 40 years in the history of automatic voice recognition (AVR).

In this paper automatic voice recognition stages and the approaches that the most of AVR systems use are shown.

Key words

Speech recognition, digital processing of signals.

Introdução

É cada vez maior a necessidade de o homem interagir de forma mais natural com as máquinas. A comunicação entre os seres humanos é feita de diversas maneiras, entre elas gestos, textos impressos, desenhos e voz. Entretanto, a voz certamente é um modo comum e natural de comunicação mais freqüentemente utilizado entre as pessoas.

O desenvolvimento de sistemas para a comunicação homem-máquina, através da voz, constitui uma área de pesquisa de grande potencial. Entretanto, a meta ambiciosa de dotar uma máquina com a capacidade de falar e entender, ainda, encontra-se distante. O avanço atual deve-se aos esforços dos pesquisadores através de um amplo espectro da ciência e tecnologia; o progresso futuro requer ainda uma relação maior entre os diversos campos de conhecimento, tais como, psicologia, lingüística, acústica, processamento de sinais, ciência da computação e microeletrônica [SCH 94].

Reconhecimento automático de voz (RAV)

Um sistema de reconhecimento automático de voz (RAV) é um sistema capaz de transformar um sinal de voz em uma seqüência de dados com a qual uma máquina tomará decisões. Os sistemas de RAV executam três tarefas básicas, assim como mostrado na Figura 1: a primeira tarefa é o **pré-processamento**, que inclui a conversão analógico/digital, filtragem e extração de parâmetros característicos do sinal de voz; a seguir, é executada uma segunda tarefa: a **identificação** da informação contida no sinal de voz; finalmente, a terceira tarefa é a **comunicação**, que consiste no envio da informação reconhecida à aplicação que tomará decisões.

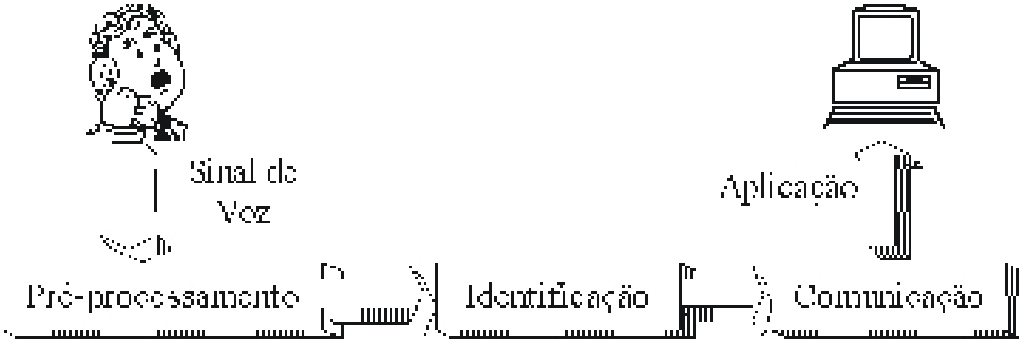


Figura 1 - Tarefas de um sistema de RAV.

A aplicação da tecnologia dos Sistemas de *RAV* pode visar a melhoria de serviços já existentes ou a oferta de novos serviços à população. Várias aplicações poderiam ser mencionadas: em linhas de produção, para descartar peças falhadas através de comandos vocálicos que movimentam uma esteira de produção; em caixas bancários automáticos, para solicitar saldos, extratos; operações por telefone, como consultas de saldos, de notas em colégio; no preparo de relatórios médicos ou odontológicos, durante ou após a consulta; em companhias aéreas, para fazer reserva de vôos; em sistemas de segurança, que identifiquem a pessoa pela voz.

O PROBLEMA DO RECONHECIMENTO DE VOZ

O *RAV* de **palavras isoladas**, em que as palavras estão separadas por uma pausa, é uma tarefa menos difícil do que o *RAV* de **fala contínua**, no qual os limites das palavras não são facilmente distinguíveis; há confusão adicional entre palavras e frases, o espaço de busca é maior e a produção de fonemas e palavras é afetada pelos fonemas e palavras vizinhos (**coarticulação**) [TEB 95].

Vocabulários grandes requerem uma capacidade de armazenamento e custo computacional também grandes. Uma solução, é não utilizar modelos de palavras, mas modelos de sub-palavras

(fonemas, grupos de fonemas ou sílabas) [MAR 96].

O *RAV* **dependente do locutor**, de boa precisão, requer treinamento para cada novo locutor, sendo inconveniente em certas aplicações. Em contraste, o *RAV* **independente do locutor**, que reconhece a voz de usuários para os quais não houve treinamento, tem precisão menor, pois os parâmetros característicos da voz dependem fortemente do usuário (idade, sexo, sotaque, velocidade da fala, etc.).

O *RAV* também é afetado por ruído; distorções acústicas; tipo, direcionamento e posição do microfone/telefone. As restrições na seqüência de palavras, também, afetam o desempenho do *RAV*, sendo representadas por uma **gramática**, que serve como filtro, com a finalidade de avaliar unicamente as sentenças possíveis [PIC 93].

Outro desafio no *RAV* é a variabilidade da voz em um único locutor e, também, para locutores diferentes. Existem duas classes de variabilidade: **acústica**, referente aos diversos acentos, pronunciações, entonações, volumes; **temporal**, mais fácil de controlar, referente às diversas velocidades de fala [TEB 95].

Pré-processamento do sinal de voz

O pré-processamento do sinal de voz, que visa gerar um conjunto de parâmetros que contêm informações relevantes à diferenciação de seus eventos, é o primeiro passo do processo de *RAV*. O pré-processamento do sinal de voz pode ser dividido em 3 sub-módulos, assim como mostrado na Figura 2. Estes sub-módulos são: (1) Captura e conversão *A/D*; (2) Análise espectral e (3) Extração de parâmetros.

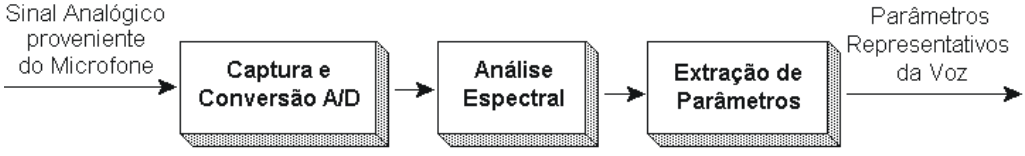


Figura 2 - Etapas do pré-processamento do sinal de voz.

CAPTURA E CONVERSÃO A/D DO SINAL DE VOZ

A conversão *A/D* consiste na amostragem do sinal analógico, $S_a(t)$, cada *T* segundos e a quantização das amostras, para obter o sinal digital $S(n)=S_a(n.T)$, *n=0, 1,...* Dada uma onda analógica contínua de largura de banda finita (onde a máxima freqüência presente é *fmax*), a informação de entrada somente será preservada com uma freqüência de amostragem *fs > 2.fmax* (critério de *Nyquist*). Para evitar o *aliasing* [RAB 93], o sinal de voz resultante do transdutor é, em uma primeira etapa, passado através de um filtro analógico, passa-baixas, com freqüência de corte $f_c \leq fs/2$, que elimina freqüências acima da metade da freqüência de amostragem. Posteriormente, o sinal passa por um conversor *A/D*.

O sinal de voz digitalizado passa através de um **filtro de pré-ênfase** com função de transferência:

$$H_{pre}(z) = 1 - a_{pre}.z^{-1} \qquad (1)$$

onde 0,9 ≤ apre ≤ 1 é o coeficiente de pré-ênfase [RAB 93]. Este filtro é utilizado para equalizar o espectro de voz, melhorar o desempenho da análise espectral, que constitui a etapa posterior [PIC 93]. A saída, do filtro de pré-ênfase, está relacionada com a entrada, S(n), através da equação:

$$S_p(n) = S(n) - a_{pre} \cdot S(n-1) \qquad (2)$$

As tarefas mencionadas, são esquematizadas na Figura 3.

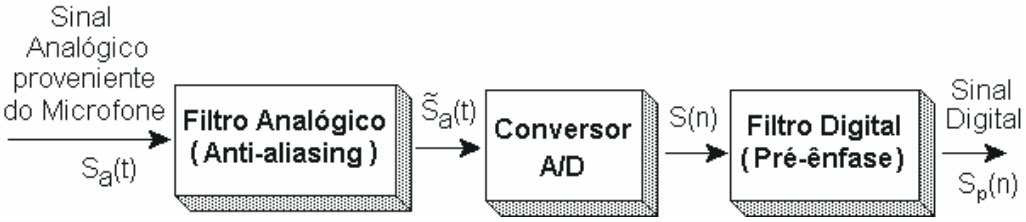


Figura 3 - Seqüência de operações na conversão do sinal analógico de voz em um sinal digital.

ANÁLISE ESPECTRAL

A Figura 4 ilustra os passos necessários para a realização da análise espectral. Inicialmente, o sinal digital é dividido em quadros de duração fixa (10ms~30ms), T_f. A análise espectral se realiza sobre um intervalo de tempo T_w (duração da janela), existindo geralmente superposição entre janelas adjacentes ($T_w > T_f$) com uma porcentagem de superposição, $\%S = 100.(T_w - T_f)/T_w$, que varia de 0% a 70% na maioria dos sistemas. Uma *%S* alta permite uma transição mais suave dos parâmetros extraídos [NUN 96]. Porém, estimativas excessivamente suavizadas podem ocultar variações reais do sinal de voz. As descontinuidades, no início e final de cada quadro, são minimizadas aplicando janelas com bordas suaves (*Hamming*, *Kaiser,* etc.) [RAB 93] [GOM 01].

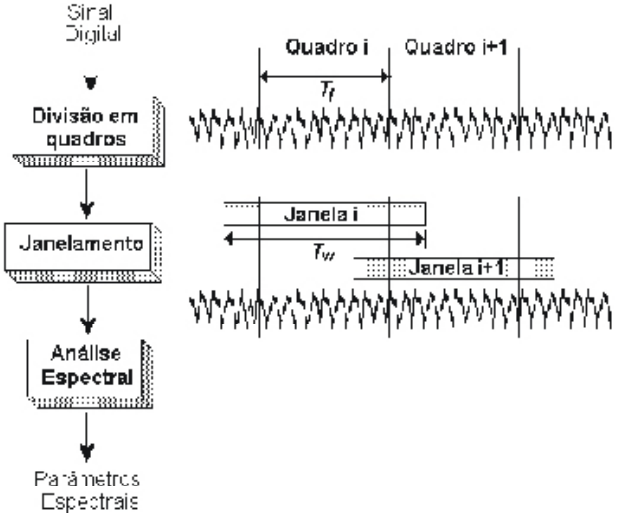


Figura 4 - Divisão em quadros e transposição entre janelas adjacentes.

Nos sistemas de *RAV* atuais, predominam dois métodos de análise espectral: o **banco de filtros** a partir da *FFT* (***F**ast **F**ourier **T**ransform*) e a **codificação preditiva linear** *(LPC* ou ***L**inear **P**redictive **C**oding*). Os espectros resultantes de ambos métodos são representações mais correlacionadas ao processo de audição e percepção humana que a representação temporal inicial, o que justifica a utilização de parâmetros extraídos a partir de tais representações espectrais [DEL 93].

Freqüentemente, realizam-se mapeamentos de freqüências que utilizam uma escala não uniforme, com significado perceptual [PIC 93], como por exemplo a escala ***Mel***, que pode ser obtida para uma determinada freqüência através da seguinte expressão:

$$f_{mel} = 2595 . log_{10}(1 + f/700) \qquad (3)$$

onde *fmel* é a freqüência (real) em *mels*.

EXTRAÇÃO DE PARÂMETROS

Após a realização de uma análise espectral do sinal de voz, é preciso determinar um conjunto de parâmetros capaz de diferenciar eventos da voz. A seleção da melhor representação paramétrica dos dados acústicos é uma tarefa importante no projeto de qualquer sistema de RAV. O objetivo fundamental, ao escolher uma representação paramétrica, é comprimir os dados de voz eliminando informação não pertinente à análise fonética dos dados e salientar aqueles aspectos do sinal que contribuem significativamente à detecção das diferenças fonéticas [RAB 93]. Os parâmetros extraídos do sinal de voz devem ser o máximo possível invariantes ao canal de transmissão, ruído de fundo, transdutor e locutor, reduzindo as taxas computacionais sem muita perda de informação [DEL 93]. A Figura 5 mostra como obter os parâmetros mel-cepstrais, através de uma análise espectral com um banco de filtros linearmente espaçados na escala mel.

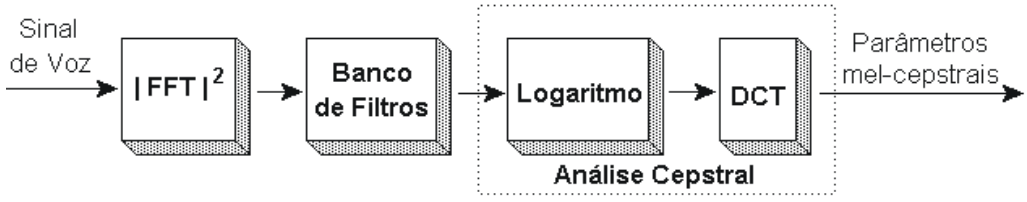


Figura 5 - Cálculo dos parâmetros mel-cepstrais baseados em banco de filtros.

É calculado o espectro do sinal de voz que foi separado em quadros através de janelas. O espectro resultante é passado através se um banco de filtros. A Figura 6 mostra tal banco de filtros passa-banda, de resposta triangular com espaçamento e largura determinados por um intervalo constante de freqüência *mel*. Calculam-se os logaritmos da energia nas saídas dos filtros e a Transformada Co-seno Discreta (*DCT*) de tais valores, obtendo os parâmetros desejados c^*_n:

$$c_n^* = \sum_{k=1}^{K} (log(S_k^*) \cdot cos\left[n \cdot \left(k - \frac{1}{2} \right) \cdot \frac{\pi}{K} \right]_{, n=1,..., L} \quad (4)$$

onde *L* é o número de coeficientes, $S^*_k(\omega)$, *k=1,...,K*, são os coeficientes de potência da saída do k-ésimo filtro e $S(\omega)$ é a entrada.

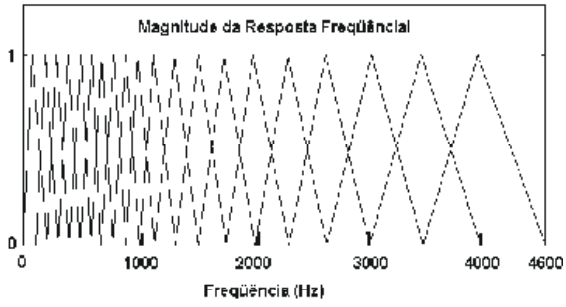


Figura 6 - Banco de filtros triangulares [RAB 93].

O número de parâmetros utilizados não necessariamente é igual ao número de filtros do banco de filtros. Freqüentemente, são descartadas algumas das últimas amostras obtidas na saída da DCT, as quais contém pouca informação sobre a forma do trato vocal utilizada para produzir o trecho de voz em análise [DEL 93].

CODIFICAÇÃO DO SINAL DE VOZ

Uma vez extraídos os parâmetros do sinal de voz, procede-se à redução da informação, utilizando algum método de codificação. A quantização vetorial (*VQ*) é uma técnica de compressão de dados que mapeia um conjunto de vetores de observação $\mathbf{X}=\{\mathbf{x}_t \in \Re^n / \ i=1...T\}$, gerando um conjunto $\mathbf{C}=\{\mathbf{y}_i \in \Re^n / \ i=1...N\}$ de *N* vetores-código. O conjunto **C** é o dicionário (*codebook*) do quantizador. O mapeamento é realizado calculando a distorção, entre cada vetor de observação $\mathbf{x}_t \in \mathbf{X}$ e os vetores-código $\mathbf{y}_i \in \mathbf{C}$, codificando $\mathbf{x}_t \in \mathbf{X}$ pelo índice *i* do vetor-código $\mathbf{y}_i \in \mathbf{C}$ que apresenta a menor distorção [MAK 85]. O objetivo do treinamento na *VQ* é gerar o melhor conjunto de vetores-código **C** que produza a menor distorção. Para tanto, existem diversos algoritmos.

A Figura 7 mostra uma variante do algoritmo *LBG* (*Linde-Buzo-Gray*) [LEE 89]. Este algoritmo separa os dados de treinamento em *2,4,...,N* partições, com os centróides das partições, representando os vetores-código desejados.

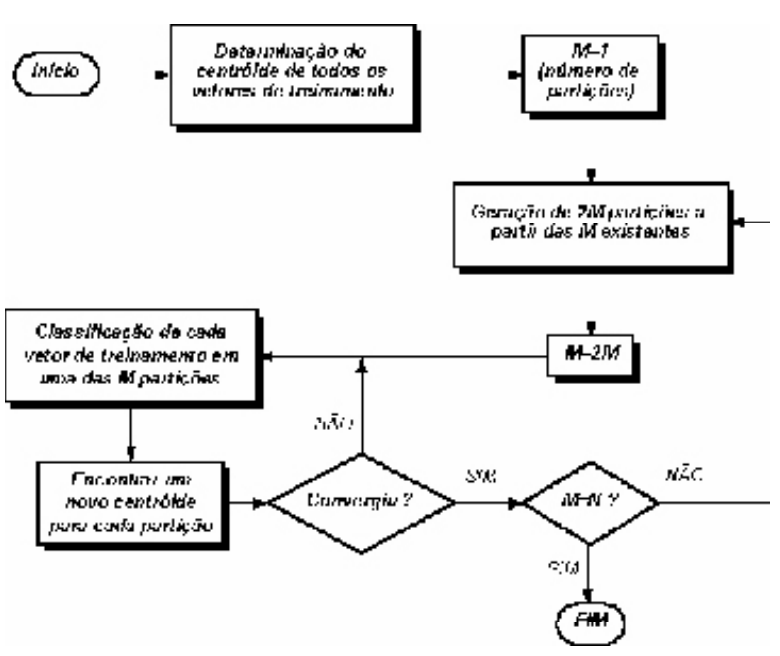


Figura 7 - Fluxograma do algoritmo LBG.

A partir de um conjunto de *M* vetores-código $\boldsymbol{V}_i^M$, *i=1,...,M*, que represente uma partição dos vetores de treinamento, obtém-se *2M* vetores $\boldsymbol{V}_i^{2M}$, *i=1,...,2M*, mediante algum método heurístico como:

$$V_i^{2M} = V_i^M + 0{,}01 \cdot V_i^M \quad , \qquad i = 1,...,M$$
$$V_i^{2M} = V_i^M - 0{,}01 \cdot V_i^M \quad , \qquad i = M+1,...,2M \qquad (\ 5\ )$$

Determinam-se, então, as *2M* partições e seus centróides através de refinamento iterativo: cada vetor de treinamento é classificado como pertencente à partição cujo centróide apresenta a menor distorção em relação a este vetor. Em seguida, calcula-se um novo centróide para cada partição através da média de todos os vetores de treinamento correspondentes. A inicialização do algoritmo (*M=1*) é realizada, calculando o centróide dos vetores de treinamento. O algoritmo termina quando o conjunto de treinamento é dividido em *N* partições, e os *N* centróides correspondentes convergem, sendo armazenados como os vetores-código do *dicionário* desejado.

Identificação de padrões de voz

Uma vez realizado o pré-processamento da voz, procedente de um determinado usuário, o sistema de *RAV* está preparado para realizar sua função primária: identificar aquilo que o usuário tenha dito. Basicamente, três grandes métodos são utilizados em *RAV* [RAB 93] e serão descritos a seguir.

MODELAGEM DETERMINÍSTICA – DTW

Esta modelagem utiliza os parâmetros do sinal de voz como padrões de teste e de referência, junto com uma métrica apropriada para a comparação dos mesmos, tendo baixo custo [RAB 93]. Existe uma diferença temporal entre ambos padrões, causada pela compressão ou expansão de fonemas e pausas dentro das frases, como mostrado na Figura 8. Isto requer um alinhamento dos padrões, utilizando algoritmos de Ajuste Temporal Dinâmico (DTW) (ver Figura 9).

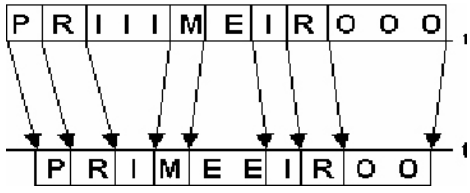
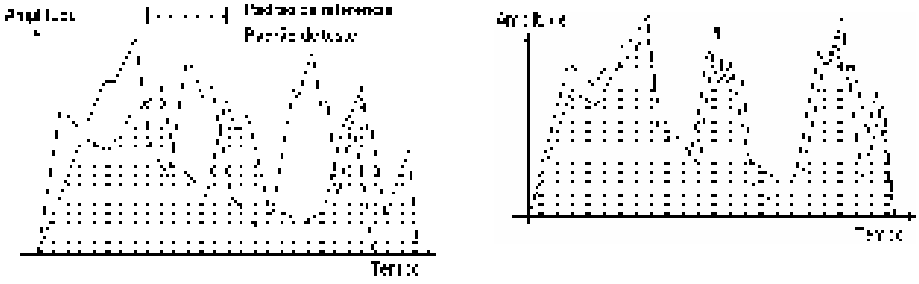


Figura 8 - Representação da diferença temporal.



(a) Antes do *DTW.* (b) Depois do *DTW.*

Figura 9 - Ajuste temporal dinâmico.

Considerando-se dois padrões de voz **R** e **T** com *N* e *M* quadros cada, correspondendo ao padrão de referência e ao padrão de teste, respectivamente, os padrões, **R** e **T** podem ser representados, depois de uma adequada extração de parâmetros, por seqüências espectrais, da seguinte maneira:

$$\mathbf{R} = \mathbf{r}_1, \ldots, \mathbf{r}_N,$$
$$\mathbf{T} = \mathbf{t}_1, \ldots, \mathbf{t}_M \qquad (6)$$

onde $\mathbf{r}_n$ e $\mathbf{t}_m$, *n=1,...,N* e *m=1,...,M*, são vetores de parâmetros de características acústicas. Os quadros de **R** e **T** são representados em um plano n-m, similar ao mostrado na Figura 10, e as diferenças temporais são descritas por uma seqüência de pontos, ou **função de ajuste**, *(n(k),m(k))* onde *k=1,...,K*, sendo *K* o comprimento do caminho.

O problema que o *DTW* deve resolver é o de encontrar o caminho, parametrizado pelo par *(n(k),m(k))*, que minimize uma distância dada. Se $D_a(n,m)$ é a distância total mínima ao longo de qualquer caminho, partindo do ponto *(1,1)* até o ponto *(n,m)*, então:

$$D_a(n,m) = \min_{n',m'} \{ D_a(n',m') + d^*[(n',m'),(n,m)] \} \qquad (7)$$

onde $d^*[(n',m'),(n,m)]$ é a distância do ponto *(n´,m´)* ao ponto *(n,m)*.

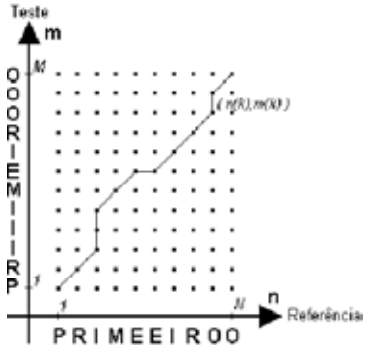


Figura 10 - Representação do DTW.

Visto que cada palavra deve ter modelo próprio, a preparação de modelos e o casamento de padrões torna-se pouco prático ao aumentar o tamanho do vocabulário. Bons resultados são obtidos com vocabulários pequenos [RAB 93].

MODELAGEM CONEXIONISTA - ANN

Atualmente, grandes esforços na área de reconhecimento de voz têm sido feitos com base em uma das técnicas emergentes da área de Inteligência Artificial: as Redes Neurais Artificiais (*ANN*). Existem centenas de arquiteturas de redes neurais propostas para aplicações que abrangem um grande número de áreas do conhecimento humano.

As redes neurais baseiam-se no funcionamento do próprio cérebro humano. Assim, possuem elementos de processamento que são modelos simplificados dos neurônios biológicos e simulam o funcionamento paralelo destes, em nosso cérebro, adquirindo características que os computadores seqüenciais carecem (paralelismo, capacidade de aprender, tolerância à falhas, etc.) [BEN 96].

O *RAV* é uma tarefa na qual uma estrutura exata dos padrões de voz é difícil de ser definida face a grande variabilidade existente, provocada pela entonação, timbre, sotaque entre outros fatores. As redes neurais podem oferecer grande auxílio na busca da solução para o problema de *RAV*. Características como a capacidade de generalização são importantes para um problema que possui como objeto de estudo algo que varia enormemente, como o sinal de voz [BEN 96].

Redes neurais são processadores paralelos que utilizam como unidades básicas neurônios artificiais. O modelo matemático do neurônio pode ser representado pela Figura 11.

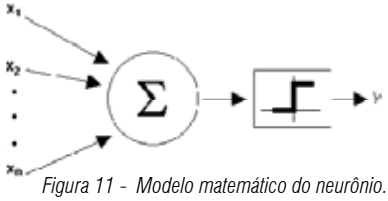


Figura 11 - Modelo matemático do neurônio.

Cada neurônio calcula a soma ponderada de n sinais de entrada *xj, j=1,...,n* e aplica esta soma em uma função. O resultado desta operação será a saída da unidade de processamento que, dependendo do tipo de rede, poderá ser discreta ou contínua. Matematicamente, a saída, *y*, da unidade de processamento será:

$$y = f\left(\sum_{j=1}^{n} w_j * x_j\right) \qquad (8)$$

onde a função *f* é conhecida como função de ativação do neurônio. Geralmente são utilizadas como funções de ativação: funções degrau, linear, sigmóide e gaussiana. Estas funções são mostradas na Figura 12.

Figura 12 - Exemplos de funções de ativação.

Na solução de problemas práticos se requererá mais de um neurônio. Uma rede neural consiste em um conjunto de neurônios arranjados, formando diversas camadas e interligados entre si. As redes neurais possuem três tipos de camadas: uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. A Figura 13 ilustra essa arquitetura básica. A concatenação de um número variado das unidades de processamento em camadas, bem como as diferentes maneiras de como serão interligadas, darão origem a um número variado de arquiteturas com características e funcionamento diferentes.

As redes neurais artificiais convencionais têm sido estruturadas para serem utilizadas em padrões estáticos [BEN 96]. Um dos grandes problemas encontrados, ao se utilizar redes neurais nos sistemas de *RAV*, é que a voz possui uma natureza dinâmica e a compressão ou a expansão que existe nas várias repetições das mesmas palavras produz seqüências de vetores de diferentes comprimentos. Como o número de unidades de entrada em uma rede neural é fixo, são necessárias algumas modificações às estruturas básicas de redes neurais. Uma forma de fazer isto é definindo um número fixo de janelas e variar o avanço e superposição entre janelas adjacentes de maneira que o número de vetores calculados sempre coincida com o número de unidades de entrada da rede neural [OGL 90] [BEN 96]. Uma outra maneira é fazer a superposição entre janelas fixas e variar o tamanho de cada janela de acordo com o tamanho da elocução sempre, procurando coincidir o número de vetores calculados na elocução com o número de entradas da rede neural [BEN 96]. Pode-se também utilizar um algoritmo que realize o ajuste temporal dinâmico e mapeie a elocução de teste em uma elocução de tamanho padrão para utilizar com a rede neural [BEN 96].

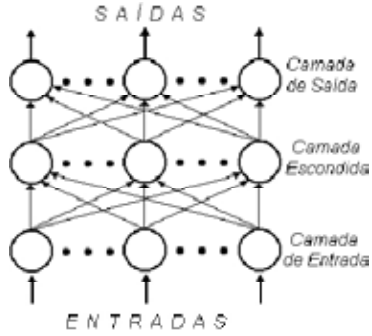


Figura 13 - Arquitetura básica de uma rede neural.

Nos sistemas de *RAV*, as redes neurais mais utilizadas tem sido: *perceptron multicamadas* ou *MLP* (**M**ulti **L**ayer **P**erceptron), *Hopfield*, *ART*, *SOM*, e em especial a rede *TDNN* (**T**ime **D**elay **N**eural **N**etwork) a qual incorpora a dinâmica do padrão de voz. Além dessas redes neurais, existem arquiteturas híbridas que se relacionam com outras técnicas tais como *DTW* e *HMM*.

MODELOS OCULTOS DE MARKOV

Um modelo oculto de *Markov* é uma máquina de estados finita estocástica, constituída por um conjunto de estados ligados entre si por transições. Ao acontecer uma transição de estado, gera-se um símbolo. Um exemplo de um *HMM* com 3 estados é mostrado na Figura 14.

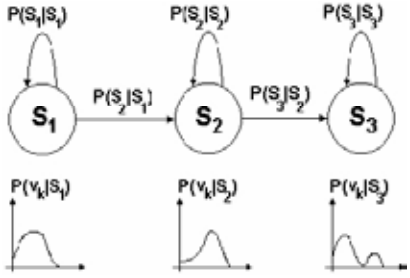


Figura 14 - HMM típico de três estados.

Um modelo de *Markov* pode ser descrito pelos seguintes parâmetros:

• Um conjunto de estados ***S={S$_i$}*** de tamanho ***N***. O estado no tempo *t* é designado como q_t. Apesar dos estados da cadeia serem ocultos, para várias aplicações existe algum significado físico para eles.

• Um conjunto de símbolos observáveis ***V={v$_k$}*** de tamanho ***M***, que corresponde à saída física do sistema que está sendo modelado.

• Uma matriz de probabilidades de transição entre estados ***A = {a$_{ij}$}***, onde a_{ij} é a probabilidade de transição do estado S_i para o estado S_j. Formalmente: $a_{ij}=P[q_{t+1}=S_j|q_t=S_i],\ 1 \le i, j \le N$.

• Um conjunto de distribuições de probabilidade. Se o *HMM* for discreto, essa distribuição é dada pela matriz ***B={b$_j$(k)}***, onde $b_j(k)$ é a probabilidade de emissão do símbolo ***k***, estando-se no estado ***j***, e pode ser definida formalmente como: $b_j(k)=P[o_t=\mathbf{v}_k|q_t=S_j],\ 1 \le j \le N;\ 1 \le k \le M$. Se o *HMM* for contínuo, será necessária a estimação das funções de distribuição de probabilidade que geram as observações dado o estado *j*, ou seja, $b_j(\mathbf{O})$ [HUA 90].

• Um vetor de probabilidades iniciais, π**={**π_i**}**, onde π_i indica a probabilidade do modelo iniciar no estado S_i. e cada π_i pode ser escrito como: $\pi_i=P[q_1=S_i],\ 1 \le i \le N$.

A notação utilizada para representar um modelo é $\lambda(\mathbf{A},\mathbf{B},\pi)$. A seqüência de observações é representada pelo vetor $\mathbf{O}=[o_1\ o_2 \ldots\ o_T]$, onde *T* é o número de observações da seqüência.

A estrutura da matriz de transição, ***A***, define o tipo de *HMM*. Um tipo de *HMM* bastante utilizado é conhecido pelo nome de modelo *left-right,* o qual tem a característica de que, ao se incrementar o tempo, o índice que indica o estado atual aumenta ou permanece igual, mas nunca diminui. A Figura 15 mostra um modelo *left-right* de 4 estados. O *HMM left-right* é apropriado para modelar sinais, tais como a voz, com propriedades variáveis no tempo de uma maneira sucessiva [LEV 83] [RAB 93].

Figura 15 - Modelo left-right.

Existem três questões básicas encontradas no desenvolvimento de sistemas modelados por *HMMs*, que devem ser resolvidas para os modelos serem úteis em aplicações do mundo real [RAB 89]. Estas questões serão descritas a seguir.

AVALIAÇÃO

O problema de avaliação resume-se em, dados um modelo $\lambda(\mathbf{A},\mathbf{B},\pi)$, e uma seqüência de observações $\mathbf{O}$, calcular a probabilidade de λ gerar a seqüência $\mathbf{O}$. A seqüência de observações $\mathbf{O}$ pode ser gerada através de diferentes seqüências $\mathbf{q}$ de estados do modelo, cada uma delas com uma determinada probabilidade. Portanto, para se calcular a probabilidade desejada, é preciso somar as probabilidades de observação da seqüência $\mathbf{O}$ para todas as seqüências de estados, de tamanho *T*, permitidas pela topologia do *HMM*.

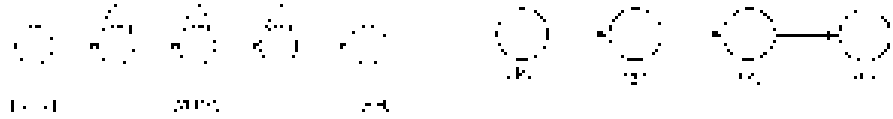
DECODIFICAÇÃO

O problema de decodificação visa a determinação da seqüência de estados $\boldsymbol{q}_i=(q_1\ q_2 \ldots\ q_T)$, do modelo λ, que mais provavelmente produziu uma dada seqüência de observações $\boldsymbol{O}$. Para determinar a seqüência de estados ótima, é necessário calcular a máxima probabilidade ao longo de um caminho, que termina no estado *i*, estando no tempo *t* e produzindo as primeiras *t* observações. Utiliza-se uma variação do algoritmo de Viterbi [RAB 89] [GOM 02].

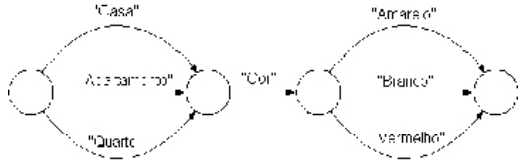
TREINAMENTO

O problema do treinamento consiste em se ajustar os parâmetros do modelo $\lambda(\boldsymbol{A},\boldsymbol{B},\pi)$, de forma a maximizar a probabilidade da seqüência de observação, $P(\boldsymbol{O}/\lambda)$. Dada qualquer seqüência de observações finita, não existe um caminho ótimo para estimar-se os parâmetros do modelo. Pode-se, porém, escolher as matrizes $\boldsymbol{A}$ e $\boldsymbol{B}$ de forma que $P(\boldsymbol{O}/\lambda)$ seja localmente maximizado. Para o treinamento, utiliza-se o algoritmo Baum-Welch (BW) [RAB 89] [GOM 01 a].

Quando os modelos de *Markov* são utilizados no reconhecimento de voz, os estados são interpretados como modelos acústicos, as distribuições de probabilidade de saída modelam os eventos de voz, enquanto que as probabilidades de transição modelam a duração destes eventos. Desta forma, um *HMM* é capaz de absorver variações temporais entre locuções de uma mesma palavra ou sentença. Um *HMM* pode ser utilizado para modelar uma elocução. Esta elocução pode ser uma sub-palavra, uma palavra, uma sentença completa ou, até mesmo, um parágrafo. Para pequenos vocabulários, normalmente se utilizam *HMMs* para modelar palavras, enquanto que para grandes vocabulários os *HMMs* são utilizados para modelar sub-palavras (fones, fonemas, etc.). A Figura 16 ilustra como os estados e as transições de um *HMM* podem ser estruturados hierarquicamente, para representar sub-palavras (neste caso, fonemas), palavras e frases [TEB 95] [FOS 98].



(a) Nível de Sub-palavras. (b) Nível de Palavras.



(c) Nível de Sentenças.

FIGURA 16 - Um modelo oculto de Markov hierarquicamente estruturado.

Utilizar modelos de palavras para grandes vocabulários pode ser proibitivo, pois o tamanho da base de dados necessário para o treinamento é proporcional ao número de palavras do vocabulário [RAB 89].

A Figura 17 mostra o diagrama de blocos de um sistema de *RAV*, baseado em *HMMs* para palavras isoladas. O diagrama assume um vocabulário de *V* palavras para ser reconhecido e que cada palavra é modelada por um *HMM* individual. Para cada palavra no vocabulário, tem-se um conjunto de treinamento de *K* repetições de cada palavra falada (falada por um ou mais locutores). Para cada palavra *v* no vocabulário, é necessário construir um *HMM* λ^v, ou seja, devem estimar-se os parâmetros de modelo $\lambda(\boldsymbol{A},\boldsymbol{B},\pi)$ que otimizem a probabilidade do conjunto de vetores de observações, utilizados para o treinamento da *v-ésima* palavra. Para cada palavra desconhecida que deve ser reconhecida, deve realizar-se a medição da seqüência de observações $\boldsymbol{O}$, através de uma análise de características da elocução correspondente à palavra, seguido pelo cálculo

das probabilidades de cada modelo $P(\mathbf{O}/\lambda^v)$, $1 \le v \le V$, e a seleção da palavra cuja probabilidade de modelo seja a mais alta:

$$v^* = \arg\max_{1 \le v \le V} \left[P(\mathbf{O} \mid \lambda^v) \right] \qquad (9)$$

O treinamento de *HMMs* em fala contínua é muito similar ao treinamento de palavras isoladas [YOU 96]. Uma das grandes vantagens do modelamento com *HMMs* é que estes podem absorver informação referente aos limites entre palavras para fala contínua [HUA 90] [TEB 95]. Visto que a seqüência de estados é oculta nos *HMMs*, então não interessa onde se encontram os limites entre as palavras. Para treinar os parâmetros dos *HMMs*, precisa-se da seqüência de palavras dentro de cada sentença. Cada palavra é instanciada com seu modelo, o qual pode ser uma concatenação de modelos de sub-palavras [MOR 95] [FOS 98]. Depois disso, as palavras na sentença são concatenadas com modelos opcionais de silêncio entre palavras. Este *HMM* concatenado da sentença é então treinado para a sentença completa. Visto que o *HMM* da sentença é treinado para a sentença completa, todos os limites das palavras estarão sendo considerados em cada quadro [LEE 89].

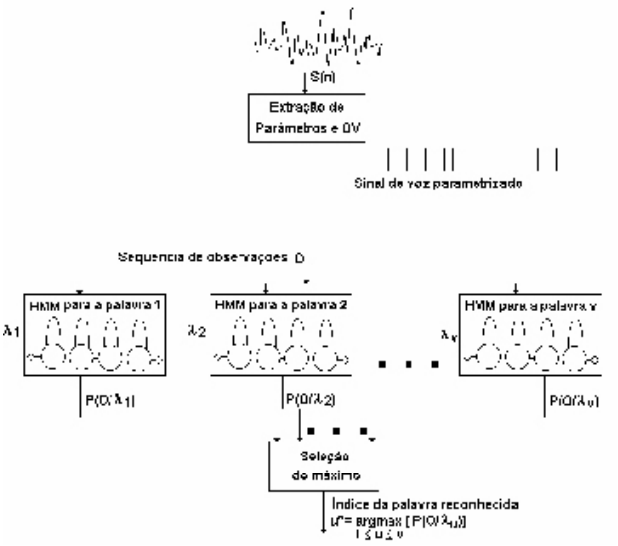


Figura 17 - Sistema de RAV para palavras isoladas sem nenhuma gramática.

Em reconhecimento de fala contínua, geralmente os limites das palavras não podem ser localizados com exatidão. Uma palavra pode começar e terminar em qualquer ponto, e é necessário levar em consideração todos os possíveis pontos iniciais e finais [YOU 96]. Isto converte a busca linear do reconhecimento de palavras isoladas em uma busca em árvore, e um algoritmo de reconhecimento polinomial em um algoritmo do tipo exponencial [LEE 89]. Porém, enquanto uma busca completa ótima é irrealizável para o reconhecimento de fala contínua com vocabulário grandes, existem varias buscas sub-ótimas que convertem o problema em um problema do tipo polinomial [LEE 89].

Atualmente, uma das mais populares soluções em RAV é a utilização da VQ na codificação de segmentos de voz junto com HMMs na modelagem e classificação de tais segmentos. O primeiro passo, a VQ, divide o espaço do sinal em um número de células para produzir um dicionário de vetores. Cada vetor no dicionário corresponde a uma célula e representa todos os vetores em tal célula. O principal objetivo da VQ é a compressão de dados, reduzindo os cálculos de processamento de sinais. As saídas da VQ representam os índices dos vetores do dicionário. O segundo passo, a modelagem HMM, produz um conjunto de modelos que representam as possíveis seqüências de vetores do dicionário que fazem parte das palavras que o sistema deve reconhecer [RAB 89].

A Figura 18 mostra um sistema, que utiliza o algoritmo LBG, mencionado na seção “Codificação do sinal de voz”. As vantagens deste algoritmo são sua simplicidade e o baixo processamento computacional. O algoritmo LBG não garante que o espaço de voz seja classificado de uma maneira globalmente otimizada. Isto quer dizer que alguns dos vetores do dicionário podem não representar as células dos vetores de entrada típicos. As limitações do algoritmo LBG conduzem a uma classificação imprecisa do espaço de voz e um casamento inadequado com a modelagem HMM. Conseqüentemente, o sistema completo produz uma precisão de reconhecimento limitada.

O algoritmo LBG é também conhecido como o algoritmo de agrupamento e divisão porque cada iteração consiste em uma etapa de agrupamento e outra de divisão. Na etapa de divisão, cada vetor gerado no dicionário na iteração precedente é dividido em dois, gerando um novo dicionário que tem duas vezes mais vetores do que o dicionário anterior. Na etapa de agrupamento, os novos vetores do dicionário são usados para agrupar todos os pontos dos dados da entrada e formar novos espaços.

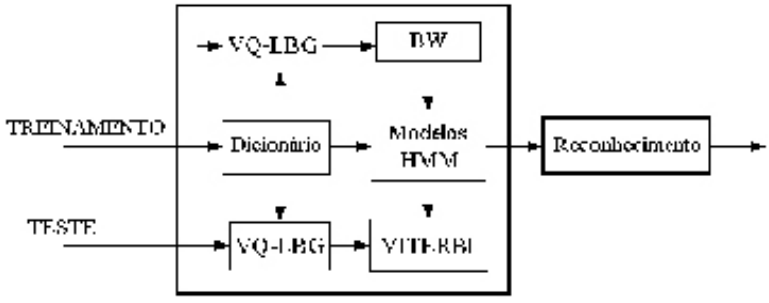


Figura 18 - Diagrama de blocos do sistema LBG/HMM.

A tabela 1 mostra a taxa de reconhecimento para a técnica LBG/DHMM. Os arquivos de voz utilizados foram gravados com ruído. Os resultados obtidos com o algoritmo LBG/DHMM são apresentados junto com os resultados do algoritmo MSVQ (Multi-Section Vector Quantization) em combinação com a regra do vizinho mais próximo (Nearest Neighbor ou NN). A taxa de reconhecimento do algoritmo LBG/HMM é menor do que a taxa obtida com MSVQ/NN. Uma implementação MSVQ/DHMM é uma alternativa que ainda não foi experimentada.

Tabela 1 - LBG/HMM vs. MSVQ/NN.

Algoritmo	LBG/DHMM	MSVQ/NN
Arquivos de Treinamento	740	740
Arquivos de Reconhecimento	370	370
Taxa de Reconhecimento	91,35%	95,95%

O RAV com DHMMs tem sido amplamente utilizado em diversas tarefas de reconhecimento, não apenas para palavras isoladas com vocabulário pequeno, mas também para reconhecimento de fala contínua tal como no caso do sistema SPHINX [LEE 89]. As principais vantagens dos DHMMs são a definição compacta e o baixo custo computacional.

Quando se utiliza um dicionário VQ para representar os vetores espectrais de voz, existe uma distorção inerente. A distorção diminui ao aumentar o nível da quantização, ou melhor dito quando o tamanho do dicionário da VQ aumenta. Isto significa que quanto maior o tamanho do dicionário, maior deveria ser a taxa de reconhecimento. No entanto, em um sistema de RAV/DHMM, isto nem sempre acontece.

A Figura 19, mostra o caso no qual se tem um vocabulário pequeno de 10 comandos para a tarefa de RAV de comandos isolados. O dicionário da VQ com 64 e com 128 palavras produz os

melhores resultados. Quando o tamanho do dicionário da VQ é aumentado para 256 ou mais, a distorção é menor do que nos casos anteriormente mencionados, mas a taxa de reconhecimento tende a piorar. Um dos motivos desta situação acontecer é que o espaço do sinal de voz consiste de grupos finitos representantes de classes acústicas, sendo que, cada classe revela algumas características estatísticas diferentes. Além disso, ao considerar as variações entre diferentes locutores e repetições do sinal de voz real, diversas palavras código poderão ser produzidas no conjunto ou na área em que há transposição de fonemas. Isto pode gerar uma VQ ambígua e piorar o desempenho do reconhecimento. Outro motivo é que o algoritmo LBG agrupa o espaço do sinal de voz geometricamente, em vez de fazê-lo estatisticamente e, assim, tem aplicabilidade limitada nos HMMs os quais produzem uma modelagem baseada nas características estatísticas da seqüência de treinamento.

Uma VQ inadequada dos dados de treinamento leva à perda de informação estatística e à eventual diminuição da precisão do sistema total de reconhecimento. Para se adequar à modelagem HMM e reduzir o efeito do erro da VQ, é necessário um método de agrupamento estatístico na primeira etapa da modelagem.

As variações em torno da média ocorrem de maneira aleatória quando é considerada uma grande população de locutores, de maneira que os pontos dentro de um grupo devem ser distribuídos segundo uma função de densidade de probabilidade Gaussiana multidimensional. Esta visão do processo de produção de voz sugere que a classificação do espaço do sinal de voz é melhor feita, utilizando um modelo de mistura Gaussiana (Gaussian Mixture Model ou GMM) nos pontos que pertencem a um determinado grupo ou "cluster" [TEB 95] [DEL 93].

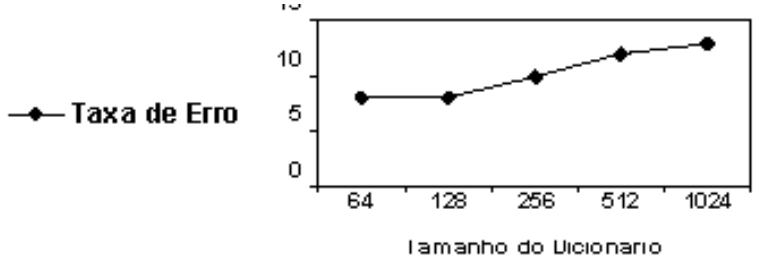


Figura 19 - Taxa de erro no reconhecimento vs. Tamanho do dicionário.

O algoritmo EM (Expectation-Maximization) pode ser utilizado como método alternativo para a VQ e o GMM, que é gerado como o dicionário da VQ. O EM é um algoritmo iterativo para a derivação da máxima verossimilhança (Maximum Likelihood ou ML) que estima os parâmetros de uma ampla variedade de modelos estatísticos e pode ser utilizado como substituto do algoritmo LBG para a quantização do espaço do sinal de voz. O algoritmo EM para GMMs é um método de quantizar o espaço do sinal de voz que reflete o processo de produção da voz de maneira mais próxima do que o algoritmo LBG. Porém, o custo computacional do algoritmo EM é maior do que o custo do algoritmo LBG.

Outro método de classificação é o algoritmo de agrupamento por gaussianas múltiplas (Multiple Gaussian Clustering ou MGC). Este algoritmo é similar ao algoritmo EM, porém, com um custo computacional menor, produzindo uma classificação bastante precisa.

A tabela 2 mostra um experimento de RAV de palavras isoladas, utilizando dígitos de zero ao nove. Tal tabela mostra que entre os sistemas de RAV individuais, o EM/HMM mostra a melhor taxa de reconhecimento, e o LBG/HMM o pior. Os três métodos de classificação dividem o espaço de dados de diferentes maneiras, e as divisões produzidas pelo algoritmo EM são as mais apropriadas.

Tabela 2 - Resultados individuais de RAV, utilizando VQ/HMM.

Tamanho do Dicionário		128	256	512
Taxa de Erro	LBG/HMM	5,67	4,86	4,13
	EMVH MM	2,75	2,67	3,16
	MGC/HMM	4,62	4,53	5,10

Uma abordagem diferente para obter uma melhor taxa de reconhecimento com VQ/HMMs é a utilização de um sistema de RAV HMM múltiplo (MHMM), baseado nos três algoritmos acima mencionados. A idéia é que outros algoritmos podem corrigir os erros produzidos pelo algoritmo individualmente. Na etapa de treinamento, os três algoritmos de classificação mencionados são utilizados para uma VQ paralela e gera-se um dicionário para cada sub-sistema. São produzidos três modelos para cada palavra do vocabulário através do algoritmo Baum-Welch. Na etapa de reconhecimento, o algoritmo de Viterbi é utilizado com os três sub-sistemas em paralelo. Tal sistema requer um maior consumo computacional e tempo de reconhecimento do que aquele que é implementado em software pelos reconhecedores individuais. Porém, no caso de um sistema em hardware [GOM 01a], os sub-sistemas poderiam ser implementados em paralelo para melhorar os resultados de reconhecimento sem incrementar o tempo de reconhecimento.

Conclusões

Realizou-se uma descrição do reconhecimento de voz, assim como as técnicas básicas que os sistemas de *RAV* utilizam. Apresentou-se a classificação dos sistemas de *RAV* e as aplicações existentes e em potencial para tais sistemas. Abordou-se o problema do *RAV*, mostrando as condições que limitam o desempenho. As tarefas dos sistemas de *RAV* foram estruturadas em três módulos: o pré-processamento, o reconhecimento e a comunicação.

A idéia de modelar, estatisticamente, as propriedades espectrais da voz fornecem uma nova dimensão do problema. A hipótese básica do modelamento estatístico é que a voz pode ser caracterizada adequadamente como um processo estocástico cujos parâmetros podem ser estimados apropriadamente.

Descreveu-se o método estatístico mais amplamente utilizado para o reconhecimento de voz, os modelos ocultos de Markov (HMM). Os HMMs estão baseados numa teoria estatística flexível que permite construir sistemas grandes de maneira consistente [FRI 96]. Os HMMs possuem uma rica representação tal que as variabilidades, no tempo e no espaço acústico, podem ser modeladas efetivamente.

As taxas de reconhecimento, ao utilizar HMMs com palavras isoladas, são de maneira geral, inferiores às obtidas quando se utilizam técnicas, tais como redes neurais e DTW. Contudo, os HMMs são muito utilizados, por diversas razões. Um motivo é que os HMMs oferecem a possibilidade de ser utilizados em sistemas híbridos HMM/ANN com os quais se obtêm melhores resultados do que com as técnicas anteriormente consideradas. Por exemplo, foi obtida uma taxa de 97,71% para o reconhecimento independente do locutor, quando se utiliza uma abordagem híbrida HMM/ANN e vozes gravadas em um canal com ruído [SCH 00].

Outra característica dos HMMs é que têm um menor consumo de memória em relação as outras técnicas. Nos HMMs, utilizam-se um conjunto de modelos de uma estrutura particular e um conjunto grande de dados de treinamento para extrair os parâmetros de cada HMM. No caso de um HMM de N estados, com M símbolos finitos por estado, é necessário estimar um total de $N^2+N.M$ parâmetros [RAB 93] [GOM 01a]. Quando N=8 e M=64, serão necessários 576 parâmetros. Estes 576 parâmetros representarão uma palavra de um determinado vocabulário, independente do número de quadros e do número de parâmetros acústicos de uma palavra. O processo de treinamento dos HMMs tem um custo computacional maior do que no caso de DTW e redes neurais, mas é um processo que precisa ser feito apenas uma única vez. Depois de ter

feito o treinamento, o processo para reconhecer palavras torna-se muito mais simples, no caso dos HMMs.

Uma situação diferente é observada no caso do DTW em que a memória requerida é dependente do número de padrões de cada palavra do vocabulário e do número de quadros e parâmetros acústicos utilizados. O número de dados, que devem ser armazenados em memória, seria Q.V.T.p dados, onde Q é o número de padrões por palavra, V é o número de palavras, T é o número de quadros de cada palavra e p é o número de parâmetros acústicos. Por exemplo, quando Q=12, V=10, T=40 e p=10, será necessário armazenar 48000 dados. Esta vantagem evidente dos HMMs se torna maior quando o vocabulário cresce ou quando se utiliza linguagem contínua, onde os HMMs podem modelar fonemas ou grupos de fonemas em lugar de palavras.

As técnicas DTW e redes neurais tornam-se inviáveis quando se trata de linguagem contínua, pois elas passam a consumir uma grande quantidade de recursos computacionais. Tais técnicas confrontam diversos problemas nos modelos de treinamento para fala contínua, já que os limites entre palavras não são detectáveis automaticamente e amiúde torna-se necessária uma separação manual [RAB 93] [JUA 84]. No entanto, os HMMs permitem uma modelagem diferente e de menor custo computacional para tais problemas [GOM 01a].

A palavra é a menor unidade de voz com significado semântico, sendo menos sensível às variações contextuais do que unidades menores, tais como o fonema [MAR 96] [YOU 96]. Como conseqüência, modelos de palavras, quando adequadamente treinados, freqüentemente levam aos melhores desempenhos de reconhecimento. Porém, nem sempre os modelos de palavras podem ser adequadamente treinados, o que vai se tornar mais grave na medida que o tamanho do vocabulário aumenta.

Utilizar modelos de palavras em grandes vocabulários pode ser proibitivo, pois o tamanho da base de dados de treinamento é proporcional ao número de palavras do vocabulário. É preferível construir modelos de palavras a partir de modelos de unidades menores, previamente treinados. Os fones representam uma escolha natural para tais unidades. Em virtude da existência de, aproximadamente, 40 fones no idioma português, o número de palavras (ou sentenças) necessário para o treinamento adequado de modelos de fones é da ordem de centenas mesmo para vocabulários grandes. A desvantagem de utilizar modelos de fones, porém, é a grande variação da pronúncia de um dado fonema dependendo do contexto no qual esteja inserido. Devido a isso, os sistemas baseados em modelos de fones apresentam, geralmente, um desempenho inferior do que os sistemas baseados em modelos de palavras [LEE 89].

Claramente, o reconhecimento de voz oferece vantagens óbvias em relação as outras formas de interação do homem com o computador. A possibilidade de emitir sons como comandos revolucionaria a vida das pessoas, tornando a interação muito mais rápida e com tempo de aprendizado menor. Infelizmente, a implementação de tal esquema é ambiciosa, pois apresenta alguns problemas muito difíceis, já que o reconhecimento de voz requer computadores mais potentes com sistemas muito complexos. A influência de fatores externos na precisão de reconhecimento também é importante, como por exemplo a utilização de microfones fixos (de preferência *head-mounted*), ambiente de gravação sem ruído, entre outros [FOS 93].

Especialmente no que se refere a sistemas de fala contínua de grandes vocabulários, o desempenho do ser humano na tarefa do reconhecimento de voz é ainda de uma ordem de magnitude superior que o correspondente desempenho das máquinas [PIC 95] [DES 97].

Os resultados das últimas décadas de pesquisa em tecnologias de *RAV*, embora não tenham resolvido todos os problemas existentes, têm apontado resultados promissores [LEE 89] [FRI 96]. Esses resultados mostram que uma ou mais soluções serão encontradas no futuro.

Referências Bibliográficas

[BEN 96] BENGIO, Y. **Neural Networks for Speech and Sequence Recognition**. New York: Thomson Computer Press, 1996. 167p.

[DEL 93] DELLER, J. R.; PROAKIS, J. G.; HANSEN, J. H. L. **Discrete-Time Processing of**

Speech Signals. New Jersey: Prentice Hall, 1993.908p.
[DES 97] DESHMUKH, N. Et. Al. Benchmarking Human Performance for Continuous Speech Recognition. In: IEEE SOUTHEASTCON, April 1997, Virginia, USA. **Proceedings...** Virginia: 1997. P.97-99.
[FOS 93] FOSTER, P.; SCHALK, T. B. **Speech Recognition. The Complete Practical Reference Guide**. New York: Telecom Library, Inc., 1993.377p.
[FOS 98] FOSLER-LUSSIER, E. **Markov Models and Hidden Markov Models: A Brief Tutorial**. Technical Report n.98-041. International Computer Science Institute, University of California, Berkeley, Dec.1998, 7p.
[FRI 96] FRITSCH, J. **Modular Neural Networks for Speech Recognition**. Pittsburgh: Carnegie Mellon University, 1996. Ph. D. Thesis.
[GOM 01] GÓMEZ-CIPRIANO, J. et. al. Design of Functional Blocks for Speech Recognition . In: **Symposium on Integrated Circuits Design** (SBCCI 2001), Brasilia.
[GOM 01a] GÓMEZ CIPRIANO, J. L. **Desenvolvimento de Arquitetura Para Sistemas de Reconhecimento Automático de Voz Baseados em Modelos Ocultos de Markov**. Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Nov. 2001. Tese de Doutorado.
[GOM 02] GÓMEZ CIPRIANO, J. L., NUNES, R.; BARONE, D. FPGA Hardware for Speech Recognition Using Hidden Markov Models, In: **International Conference on Speech and Language Processing (ICSLP 2002)**, Colorado.
[HUA 90] HUANG, X. D.; ARIKI, Y.; JACK, M. A. **Hidden Markov Models for Speech Recognition**. Edinburgh: Edinburgh University Press, 1990.276p.
[LEE 89] LEE, K.- F. A**utomatic Speech Recognition - The Development of the SPHINX System**. London: Kluwer Academic Publishers, 1989.207p.
[LEV 83] LEVINSON, S. E.; RABINER, L.R.; SONDHI, M. M. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. **The Bell System Technical Journal**, USA, v.62 n.4, p.1035-1074. April 1983.
[MAK 85] MAKHOUL, J. Et al. Vector Quantization in Speech Coding**. Proceedings of the IEEE**, New York, v.73, n.11, p.1551-1589, Nov., 1985.
[MAR 96] MARKOWITZ, J. A. **Using Speech Recognition**. New Jersey: Prentice-Hall, 1996.292p.
[MOR 95] MORGAN, N.; BOURLARD, H. Continuous Speech Recognition. **IEEE Signal Processing Magazine**, p.25-42. May 1995.
[PIC 93] PICONE, J.W. Signal Modeling Techniques in Speech Recognition. **Proceedings of the IEEE**, v.81, n.9, p.1215-1247. Sept.1993.
[PIC 95] PICONE, J.; EBEL, W. J.; DESKHMUKH, N. Automated Speech Understanding: The Next Generation. **Digital Signal Processing Technology**, v. CR57, p.101-114.1995
[RAB 89] RABINER, L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. **Proceedings of the IEEE**, New York, v.77, n.2, 257-286, Feb.1989.
[RAB 93] RABINER, L.; JUANG, B.- H. **Fundamentals of Speech Recognition**. New Jersey: Prentice Hall, 1993.507p.
[SCH 00] SCHRAMM, M. C. et al. A Brazilian Portuguese Language Corpus Development. In: **International Conference on Speech and Language Processing, ICSLP 2000**, Beijing. Proceedings... Beijing: [S.l.:s.n.], 2000. p.579-582.
[SCH 94] SCHAFER, R.W. Scientific Bases of Human-Machine Communication by Voice. In: **Voice Communication Between Humans e Machines**, p.15-33, 1994.
[TEB 95] TEBELSKIS, J. **Speech Recognition Using Neural Networks**. Pittsburgh: Carnegie Mellon University, 1995. Ph. D. Thesis.
[YOU 96] YOUNG, S. A Review of Large-Vocabulary Continuous-Speech Recognition. **IEEE Signal Processing Magazine**, p.45-57, Sept.1996.