2
Reconhecimento Voz Contínua em Sistemas Distribuídos

O conceito de reconhecimento de voz distribuído (*DSR – Distributed Speech Recognition*) foi desenvolvido como uma forma eficiente de transladar a tecnologia de Reconhecimento Automático de Voz (RAV) para o ambiente móvel e redes IP.

A idéia do *DSR* consiste em usar um *front-end* local, de onde os parâmetros de voz são obtidos e transmitidos através de um canal de dados, até um *back-end* onde se localiza o reconhecedor de voz. Esta idéia pode ser observada na Fig. 2.1.

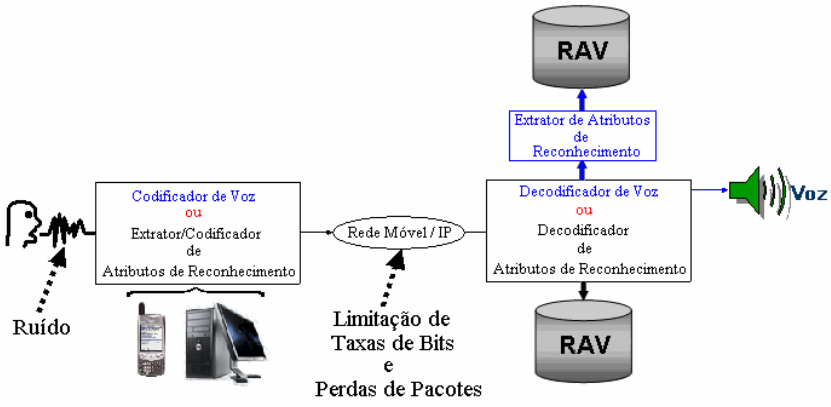

Figura 2.1 Sistemas de Reconhecimento Distribuído – Diagrama Básico

Outra característica importante desta abordagem está no fato de que uma análise relativamente simples de voz é realizada no *front-end* local, enquanto que a maior parte do processamento é colocada no servidor de reconhecimento que pode ser facilmente atualizado para novas tecnologias e serviços, sem custo adicional para o usuário [3].

2.1.
Extrator de Atributos em Sistemas Distribuídos

A partir das características básicas de um *DSR*, é importante analisar as abordagens de reconhecimento com os respectivos atributos utilizados, de forma a definir um bom sistema a ser implementado e o que nele pode ser melhorado.

Cabe ressaltar que a dedução matemática detalhada dos atributos a serem utilizados no reconhecimento será feita no Capítulo 4. Aqui serão apresentadas três abordagens diferentes e os atributos de reconhecimento mais utilizados em cada uma delas [4], [5] e [6].

A. Extrator de atributos usando os parâmetros do codificador de voz

Um esquema deste tipo é ilustrado na Fig. 2.2, onde pode ser observada a sua adequação às situações em que se queira realizar o reconhecimento e a recuperação da voz do locutor.

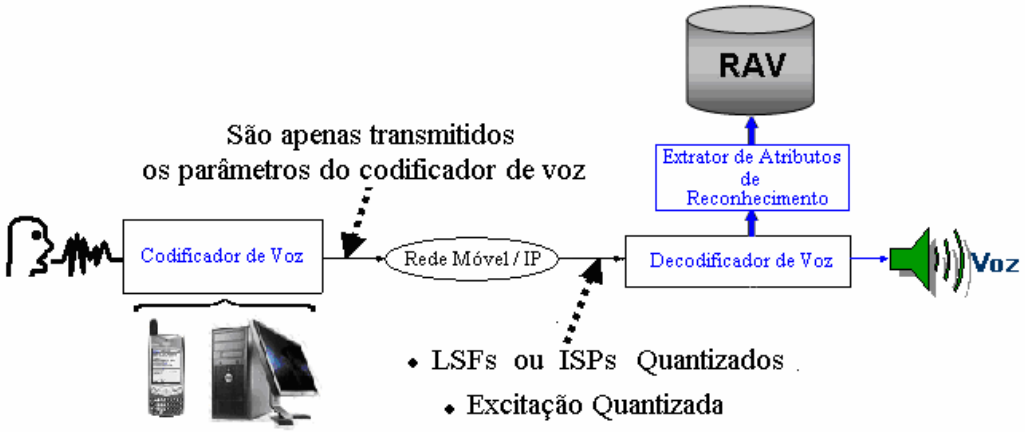

Figura 2.2 – Sistema de reconhecimento de voz distribuído baseado nos parâmetros de voz do codificador

Neste sistema existe uma ampla variedade de atributos de reconhecimento que podem ser obtidos a partir dos parâmetros do decodificador no processo de recuperação da voz, o que simplifica bastante o extrator de atributos.

O codificador de voz transmite as LSFs ou os ISFs e os parâmetros da excitação quantizados, como apresentado na Fig. 2.2. Estes parâmetros da voz trafegam pela rede e chegam ao receptor, onde deseja-se realizar o reconhecimento.

Diretamente dos parâmetros LSFs quantizados pode-se obter atributos de reconhecimento, sendo eles em número de quatro: PCC (*Pseudo-Cepstral Coefficients*) [1], PCEP (*Pseudo-Cepstrum*) [4], MPCC (*Mel Pseudo-Cepstral Coefficients*) [1] e MPCEP (*Mel Pseudo-Cepstrum*) [4].

Dos parâmetros ISFs quantizados tem que se investigar o desempenho quando utilizados diretamente no reconhecimento, bem como possíveis transformações para atributos de reconhecimento.

Dos parâmetros LSF quantizados ou dos ISF quantizados o decodificador de voz do receptor obtém os parâmetros LPC de onde se podem extrair dois atributos de reconhecimento: LPCC (*LPC Cepstrum*) [5] e MLPCC (*Mel LPC Cepstral Coefficients*) [5].

Seria interessante também neste cenário ser analisada a combinação da informação da excitação com os atributos de reconhecimento obtidos de LSF, ISF ou LPC de forma a obter um novo vetor de atributos que tenha melhor desempenho que os originais.

B. Extrator de atributos a partir de voz decodificada

Uma ilustração deste sistema é apresentada na Fig. 2.3, onde se pode observar que o mesmo tem que recuperar a voz do locutor, para efetuar o reconhecimento, o que tem demonstrado desempenho inferior ao das demais abordagens [6].

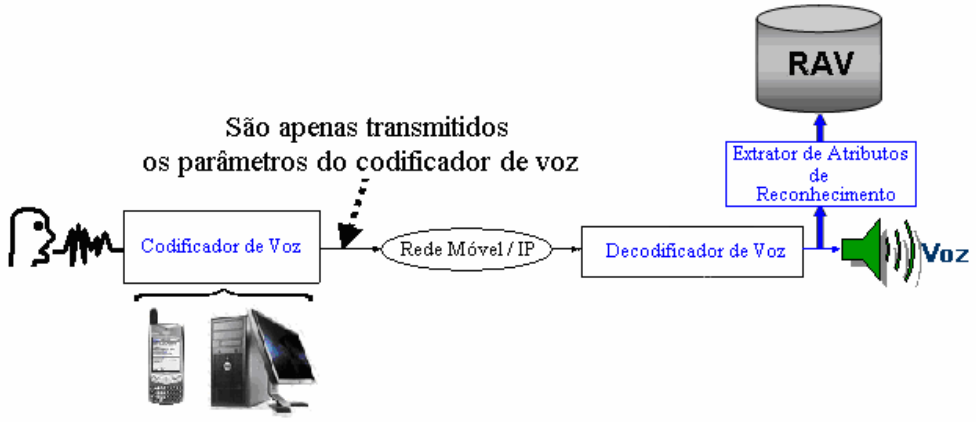

Figura 2.3 – Sistema de reconhecimento de voz distribuído baseado em voz decodificada

Da voz reconstruída podem ser obtidos vários atributos de reconhecimento, dentre os quais: MFCC (*Mel-Frequency Cepstral Coefficients*) [7], AMFCC (*Autocorrelation Mel-Frequency Cepstral Coefficients*) [8], RAS (*Relative Autocorrelation Sequence*) [9], ZCPA (*Zero Crossings with Peak Amplitudes*) [10], PS-ZCPA (*Pitch-Synchronous Zero Crossings with Peak Amplitudes*) [11],

PS-PA (*Pitch-Synchronous Peak Amplitude*) [12], SSCH (*Subband Spectral Centroid Histograms*) [13].

C. Extrator de atributos para reconhecimento no terminal do usuário

Uma ilustração deste sistema é apresentada na Fig. 2.4, onde se pode observar que o mesmo é bastante adequado para situações onde se deseja realizar apenas o reconhecimento, não sendo um requerimento a recuperação de voz.

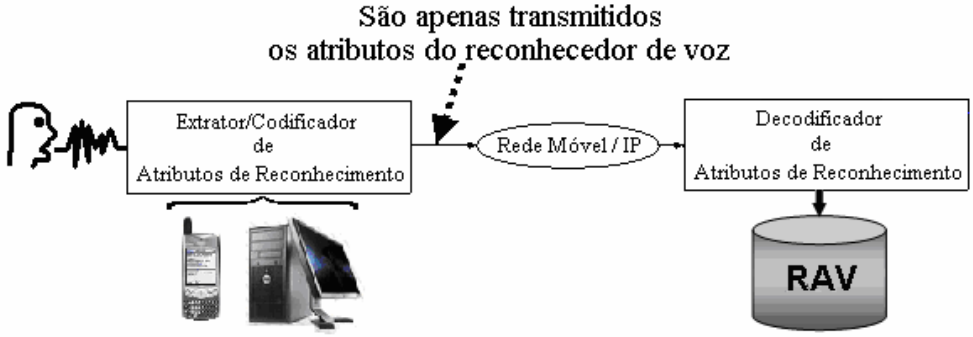

Figura 2.4 – Sistema de reconhecimento de voz distribuído com codificação dos atributos de reconhecimento no *front-end* local

Este sistema pode ser combinado com um sistema de codificação de voz, porém, isto implicará em maior quantidade de informação a ser transmitida no canal e maior processamento do *front-end* local.

Os atributos de reconhecimento gerados pelo codificador de atributos do *front-end* local são os mesmos obtidos de voz reconstruída apresentados no reconhecedor de voz, a partir de voz decodificada. No entanto, os mesmos são obtidos de voz original. Estes sistemas apresentam desempenho igual aos sistemas de reconhecimento automático de voz onde todo o processamento é feito em um único local [6].

2.2. Reconhecimento de Voz Contínua

Sistemas atuais de reconhecimento de voz contínua (*CSR – Continuous Speech Recognition*) com amplo vocabulário (*LVR – Large Vocabulary Recognition*) são estritamente baseados nos princípios de reconhecimento

estatístico de padrões. Os métodos básicos em que se aplicam estes princípios têm ainda forte influência de sistemas pioneiros da década de 70 [14], [15]. A arquitetura representada na Fig. 2.5 é praticamente um consenso na área e é composta pelos seguintes componentes: modelos acústicos, léxico de palavras (opcional), modelo linguístico e, principalmente, o decodificador. Estes blocos serão explorados no restante desta seção.

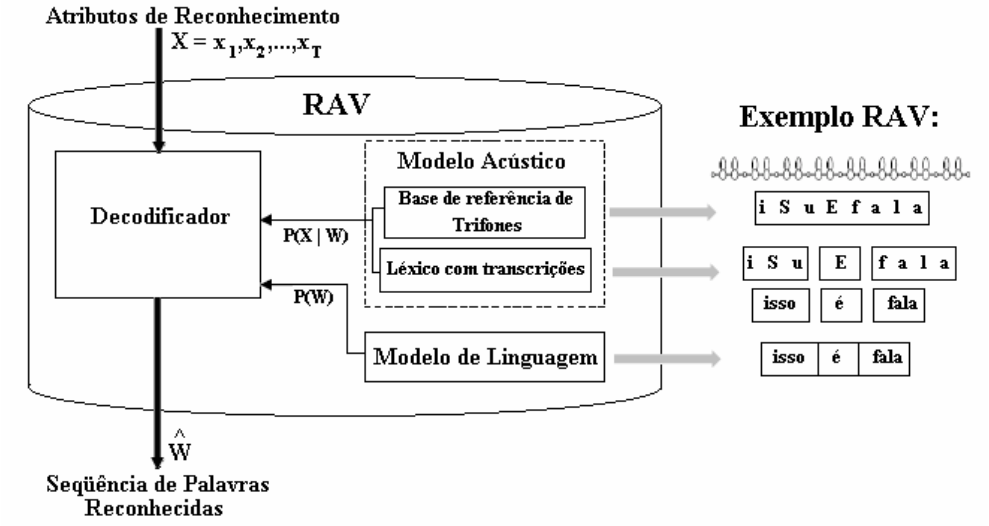

Figura 2.5 – Diagrama em blocos de um sistema de reconhecimento automático de voz baseado em modelos estatísticos de subunidades de palavras [16]

Um sistema de reconhecimento de voz converte o sinal acústico observado em sua representação ortográfica correspondente. Houve um grande avanço na resolução deste problema através da utilização de um modelo estatístico de distribuição conjunta, $P(W,X)$, entre a sequência de palavras pronunciadas W e a sequência de informações acústicas observadas X, numa abordagem conhecida como modelo fonte canal [17]. Nesta abordagem, o sistema de reconhecimento procura uma estimativa $\hat{W}$, da sequência de palavras pronunciadas, a partir da evidência acústica observada X, usando a distribuição de probabilidades a posteriori, $P(W \mid X)$. Para minimizar a taxa de erro, o sistema escolhe a sequência de palavras que maximiza essa distribuição

$$\hat{W} = \arg\max_{w}\left[P(W \mid X)\right] = \arg\max_{w}\left[\frac{P(W)P(X \mid W)}{P(X)}\right] \qquad (2.1)$$

Em (2.1), após a aplicação do Teorema de Bayes, a distribuição a posteriori é decomposta em $P(W)$, a probabilidade a priori da sequência de palavras W, e $P(X|W)$ que é a probabilidade de observar a evidência acústica X quando a sequência W é pronunciada. A distribuição $P(W)$ refere-se às palavras que poderiam ter sido pronunciadas (a fonte) e está associada a um modelo de linguagem. O modelo de probabilidade de uma observação $P(X|W)$ (o canal) é chamado de modelo acústico.

A. Modelo Acústico

O propósito da modelagem acústica é prover um método que calcule a verossimilhança de qualquer sequência de vetores X, dada uma sequência de palavras W. A princípio, a distribuição de probabilidade requerida, $P(X|W)$, pode ser modelada através de inúmeras palavras e o cálculo estatístico de sequências de vetores correspondentes. No entanto, esse método é impraticável para sistemas com amplo vocabulário e, ao invés disso, as palavras modeladas pelo sistema são usualmente decompostas em seus respectivos fones.

Nesta tese, seguindo algumas das implementações com melhor desempenho no cenário de reconhecimento distribuído de voz contínua para amplo vocabulário [18], cada fone foi representado por um HMM (*Hidden Markov Model*) de primeira ordem que contém três estados emissores e uma topologia simples do tipo esquerda-direita (*left-right*). Estados de entrada e saída, não emissores, foram acrescidos à modelagem para facilitar a união entre modelos, ficando o modelo conforme ilustrado na Fig 2.6. O estado de saída (não emissor – não possui função densidade de probabilidade atrelada a este estado) do modelo de um fone pode ser unido com o estado de entrada (não emissor) de outro para criar um HMM composto. Isto permite que modelos de fone sejam unidos para formar palavras e estas unidas para formar frases completas. No entanto, o modelo de pausa, por ser estacionário, foi representado por uma topologia mais simples, constituída apenas de um estado emissor de saída.

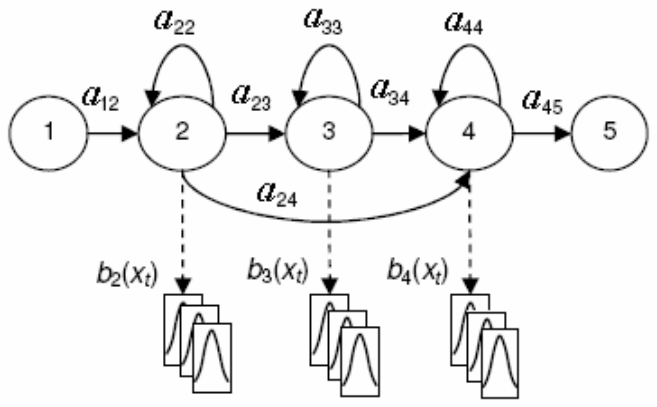

Figura 2.6 – Modelo de fone baseado em HMM

HMM é uma máquina de estados finita que modifica seu estado a cada unidade de tempo e, a cada tempo t e estado j, um vetor acústico de fala x_t é gerado com função densidade de probabilidade $b_j(x_t)$, composta nesta tese pela mistura de 20 (M) gaussianas contínuas multivariadas com covariância diagonal [18], pois em [16] formaram simuladas com outras quantidades de gaussianas (1, 4, 8, 12, 16) e a com 20 gaussianas foi a que apresentou melhor desempenho:

$$b_j(x_t) = \sum_{m=1}^{M} c_{jm}\, \eta(x_t, \mu_{jm}, \Sigma_{jm}) \tag{2.2}$$

onde c_{jm} é o peso do componente m da mistura, no estado j, e $\eta(x, \mu, \Sigma)$ denota uma Gaussiana multivariada de vetor média μ e matriz covariância Σ,

$$\eta(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)' \Sigma^{-1} (x-\mu)} \tag{2.3}$$

onde $(\cdot)'$ indica transposta e $|\cdot|$ indica determinante.

Além disso, a transição de um estado i para um estado j também é probabilística e governada por uma probabilidade discreta a_{ij}.

A função densidade de probabilidade conjunta condicionada de uma sequência de vetores X e uma sequência de estados S, dado um modelo M, é calculada pelo produto das probabilidades de transição de estados com a função

densidade de probabilidade de emissão de saída. Assim, a função densidade de probabilidade conjunta condicionada de uma sequência de vetores acústicos X e uma sequência de estados $S = s(1), s(2), s(3), ..., s(T)$ é

$$p(X, S \mid M) = a_{s(0)s(1)} \prod_{t=1}^{T} b_{s(t)}(x_t) a_{s(t)s(t+1)} \tag{2.4}$$

onde o estado inicial $s(0)$ é restringido ao estado de entrada e $s(T+1)$ é restringido ao estado de saída. Na prática, apenas a sequência observada X é conhecida, enquanto a suposta sequência S fica escondida pelo modelo. Por isto a técnica é denominada Modelos Escondidos de Markov. A função densidade de probabilidade condicionada $p(X \mid M)$, necessária para resolver (2.1), é facilmente encontrada somando a Equação (2.4) sobre todas as possibilidades de sequências de estado

$$p(X \mid M) = \sum_{s} p(X, S \mid M) \tag{2.5}$$

Um método recursivo e eficiente para realizar este cálculo é o algoritmo avanço-retorno (*Forward-Backward*). Uma característica crucial desse algoritmo é que ele também permite o cálculo da probabilidade de se estar num estado específico do modelo em um instante específico de tempo. Isso leva a um procedimento simples e eficiente conhecido como algoritmo Baum-Welch [19], o qual procura estimativas de máxima verossimilhança para o conjunto de parâmetros de cada HMM.

No exposto até aqui, assumiu-se de forma implícita ser necessário apenas um HMM por fone, porém na prática, efeitos contextuais causam uma larga variação na maneira como sons diferentes são produzidos, chamando-se de alofones os fones existentes em diferentes contextos. Logo, para se conseguir uma boa discriminação fonética, distintos HMMs devem ser utilizados para representar os fones em cada um dos diferentes contextos. Nesta tese foram utilizados trifones para resolver esse problema, onde cada um dos fones possui modelos distintos de HMM para os pares formados por fones situados à direita e a esquerda. Os contextos de trifones podem abranger ou não as fronteiras entre palavras. Quando

são consideradas as fronteiras os trifones são chamados de trifones entre-palavras (*cross-word triphones*) o que resulta em uma modelagem mais precisa e com melhor desempenho, o que será utilizado nesta tese. O contrário ocorre quando as fronteiras são desconsideradas, onde temos os chamados trifones intrapalavra (*word-internal triphones*), o que resulta em um sistema com menor habilidade em se modelar efeitos contextuais nas fronteiras de palavras e na modelagem de fala fluente, resultando em um desempenho um pouco pior no reconhecimento.

O uso de misturas de Gaussianas nas distribuições de saída permite que as distribuições de cada estado sejam modeladas com muita precisão. No entanto, quando trifones são usados, o sistema resultante requer o treino de muitos parâmetros, logo precisa de uma base muito grande para ser treinado. O problema de muitos parâmetros e poucos dados de treino é absolutamente crucial no projeto de um reconhecedor de fala estatístico.

Uma tentativa de se resolver este problema, e que será utilizado nesta tese, é a união de estados (*state-tying*) [20], [21], nesta técnica são unidos os estados que são acusticamente indistinguíveis (chamados comumente de senones). Isso permite que os dados associados a cada estado individualmente sejam combinados num recurso em comum e, assim, estimados de forma mais robusta. Após as uniões, vários estados passam a compartilhar as mesmas distribuições.

A escolha sobre quais estados devem ser unidos será realizada através de árvores de decisão fonética [22], [23], [24]. Este método envolve a construção de uma árvore binária para cada fone e posição de estado. Cada árvore tem, em cada um dos nós, uma pergunta fonética do tipo sim/não, como “O contexto à esquerda é nasal?”. Inicialmente, todos os estados de um dado fone são posicionados no nó raiz da árvore. De acordo com as respostas, o conjunto de estados é sucessivamente dividido até que os estados tenham alcançado os nós terminais da árvore. Todos os estados depositados em um mesmo nó terminal são então unidos.

As questões em cada nó são escolhidas para maximizar a verossimilhança entre os dados de treino e o conjunto resultante da união de estados. Na prática, as árvores de decisão fonética resultam em grupamentos de estados compactos e de boa qualidade, os quais possuem dados suficientes para estimar de forma robusta as misturas de Gaussianas das funções densidade de probabilidade de emissão de saída. Além disso, elas podem ser usadas para sintetizar um HMM para qualquer

possível contexto, apareça ele nos dados treino, ou não, simplesmente descendo na árvore e usando as distribuições de estado associadas aos nós terminais.

B. Modelo de linguagem

O modelo de linguagem (*LM – Language Model*) pode ser formulado como a probabilidade $P(W)$ definida por

$$P(W) = P(W_1^k) = P(w_1,...,w_k) \tag{2.6}$$

onde reescreve-se $P(W)$ como $P(W_1^k)$ acrescentando os índices 1 e k de forma a representar uma sequência com k palavras. A probabilidade conjunta de ocorrência de palavras da (2.6) pode ser substituída por um produto de suas probabilidades condicionais da seguinte forma

$$P(W_1^k) = P(w_1)\prod_{i=2}^{k} P(w_i \mid w_1,...,w_{i-1}) \tag{2.7}$$

Uma maneira simples, porém efetiva, de se obter estas probabilidades é com a utilização de n-gramas, onde se assume que w_k depende apenas das n-1 palavras anteriores a ela, reduzindo a equação (2.7) para

$$P(W_1^k) \approx P(w_1)\prod_{i=2}^{k} P(w_i \mid W_{i-(n-1)}^{i-1}) \tag{2.8}$$

Na tentativa de capturar a correlação existente entre palavras vizinhas, os n-gramas acabam simultaneamente absorvendo sintaxe, semântica e pragmática existente nas frases observadas. Isso os faz extremamente efetivos em idiomas como inglês ou português onde a ordem das palavras é importante visto que os efeitos contextuais mais fortes normalmente vêm dos vizinhos mais próximos. Além disso, as distribuições de probabilidade dos n-gramas podem ser computadas diretamente de textos prontos e, portanto, não há necessidade de se definir regras explícitas de linguística como uma gramática formal do idioma.

A princípio, os n-gramas podem ser estimados através de simples contadores de frequência e armazenados em uma tabela de memória (*look-up table*). Para o caso de trigramas (n=3), que será adotado em todos os testes desta tese [18]

$$P(w_k \mid w_{k-2}, w_{k-1}) = \frac{t(w_{k-2}, w_{k-1}, w_k)}{b(w_{k-2}, w_{k-1})} \tag{2.9}$$

onde $t(w_{k-2}, w_{k-1}, w_k)$ é o número de vezes que o trigrama w_{k-2}, w_{k-1}, w_k aparece nos dados de treino e $b(w_{k-2}, w_{k-1})$ é o número de vezes que o bigrama w_{k-2}, w_{k-1} aparece. Para lidar convenientemente com a condição de início de frase, associa-se uma nova palavra, w_0, a um símbolo de início de frase, com probabilidade $P(w_0) = 1$. Utilizando (2.7) e (2.8) tem que $P(w_1) = P(w_1 \mid w_0)$, podendo $P(w_1 \mid w_0)$ ser estimado pelo número de vezes que a palavra w_1 começou a frase dividida pelo número de frases da base utilizada para gerar o modelo de linguagem. Em seguida prossegue-se na forma indicada por (2.9) para o cálculo distribuição da probabilidade dos trigramas. Pode-se, claramente, perceber o problema de que, para um vocabulário com V palavras, existem V^3 potenciais trigramas. Para o vocabulário de 60.080 palavras desta tese (aproximadamente 10^{15} possíveis trigramas), este número é extremamente grande. O fato de que muitos dos possíveis trigramas não aparecerão nos dados de treino (240.000 frases da base de treino utilizada nesta tese extraídos de [25]) e muitos outros aparecerão apenas uma ou duas vezes, faz com que a estimativa de (2.9) seja muito pobre. Em resumo, há um problema agudo de escassez de dados.

Uma das soluções de treinamento projetada como solução à escassez de dados é o uso de uma combinação de técnicas de desconto (*discounting*) e retrocesso (*backing-off*) [26], [27]. Desconto significa que as contagens de trigramas que ocorrem com maior frequência são reduzidas e a massa probabilística resultante em excesso é redistribuída entre os trigramas que ocorrem com menos frequência, ou mesmo não ocorram. O retrocesso é aplicado quando há pouquíssimos trigramas para formar qualquer tipo de estimativa (ex.: apenas uma ou duas ocorrências nos dados de treino). Ele envolve a substituição

da probabilidade de um trigrama por uma probabilidade de bigrama escalada, que é

$$P(w_k \mid w_{k-2}, w_{k-1}) = B(w_{k-2}, w_{k-1}) P(w_k \mid w_{k-1}) \tag{2.10}$$

onde $P(w_k \mid w_{k-1})$ é a probabilidade bigrama e $B(w_{k-2}, w_{k-1})$ é um fator de escalamento calculado de forma a garantir que a soma das probabilidades trigrama some um

$$\sum_{w_{k-2}, w_{k-1}} P(w_k \mid w_{k-2}, w_{k-1}) = 1 \tag{2.11}$$

Ainda que uma estimativa robusta das probabilidades de trigramas requeira um cuidado considerável, os problemas decorrentes são solucionáveis e um bom desempenho tem sido obtido. A técnica de n-gramas possui deficiências óbvias que resultam da sua inabilidade de explorar restrições de maior amplitude como a concordância entre sujeito e verbo de uma frase [28]. Como consequência, várias alternativas foram estudadas como modelos baseados em árvore [29], modelos em treliça [30], modelos com gatilhos [31], modelos com histórico [32] e n-gramas variáveis [33]. Entretanto, em geral, todas estas tentativas resultaram em apenas pequenos ganhos de desempenho sobre um considerável custo computacional. Por esse motivo, após quase duas décadas de pesquisa, os modelos de linguagem conhecidos como bigramas e trigramas ainda dominam os sistemas *LVR*. Nestes sistemas, têm se usado, para modelagem da linguagem, bases de dados de 1 milhão a 500 milhões de palavras, correspondendo a vocabulários de 1 mil a 267 mil palavras distintas, para construção dos modelos trigrama [17] (nesta tese utilizou-se uma base de dados com 3.993.906 palavras, correspondendo a um vocabulário de 60.080 palavras distintas – sendo este o tamanho do vocabulário utilizado no decodificador – em 240.000 frases extraídas de [25]).

C. Decodificador

O termo decodificador, no contexto de reconhecimento de voz, foi criado numa analogia à terminologia usada nos métodos que utilizam estados finitos para decodificação [34] no campo da teoria da informação.

A decodificação é um processo de busca no qual uma sequência de vetores correspondentes a características acústicas (atributos) do sinal de voz é comparada com modelos de palavras. De uma maneira geral, o sinal de voz e suas transformações não fornecem uma indicação clara das fronteiras entre palavras nem do número total de palavras em uma dada locução, de modo que a determinação destas é parte do processo de decodificação. Neste processo, todos os modelos das palavras (formadas por seus respectivos modelos de fones) são comparados com uma sequência de vetores acústicos. O número de modelos cresce com o vocabulário, e pode gerar espaços de busca grandes, o que torna o processo de busca oneroso em termos computacionais, e portanto lento. Em geral, esta etapa do reconhecimento, nos sistemas modernos, é responsável por praticamente todo o esforço computacional no reconhecimento de fala contínua e, portanto, é a que determina a velocidade final desses sistemas.

Durante o processo de maximização de (2.1), repetida por conveniência a seguir, o termo $P(X|W)$ é expandido em função do modelo acústico, vinculando os estados dos HMMs à emissão de saídas nos mesmos. Dessa forma, $P(X|W)$ passa a ser calculado como a soma de todas as possibilidades de transições entre as possíveis sequências de estados do modelo sob hipótese

$$\begin{aligned}\hat{W}_1^N &= \arg\max_{W_1^N}\left\{P(W_1^N)P(X_1^T|W_1^N)\right\}\\ &= \arg\max_{W_1^N}\left\{P(W_1^N)\sum_{S_1^T}P(X_1^T,S_1^T|W_1^N)\right\}\\ &\approx \arg\max_{W_1^N}\left\{P(W_1^N)\max_{S_1^T}P(X_1^T,S_1^T|W_1^N)\right\}\end{aligned} \tag{2.12}$$

onde $W_1^N = w_1...w_N$ representa a hipótese de sequência de palavras (compostas por seus respectivos HMMs de subpalavras), $S_1^T = \{s_1...s_T\}$, a hipótese de sequência

de estados dentro do modelo e $X_1^T = \{x_1 ... x_T\}$, os vetores acústicos observados. O somatório nesta equação é então substituído por uma maximização, num processo referido como Aproximação de Viterbi [15]. Ao invés de somar sobre todos os caminhos, considera-se apenas o caminho mais provável.

Neste processo de maximização, o espaço de busca pode ser descrito como uma rede onde se busca o melhor alinhamento temporal entre a sequência de entrada e os estados dos modelos. A busca pode ser realizada em dois níveis: no nível de estados (S_1^T) e no nível de palavras (W_1^N). É possível recombinar eficientemente as hipóteses nos dois níveis usando programação dinâmica (*DP - Dynamic Programming* [35]), limitando a explosão combinatória do número de hipóteses de busca.

Nesta tese será usada a decodificação de Viterbi [16] com feixe de busca que é um algoritmo de programação dinâmica que procura no espaço de estados a mais provável sequência de estados que modele o trecho de fala de entrada. O espaço de estados é construído pelos chamados Palavra-HMMs, que são formados pela concatenação dos HMMs dos trifones que as constituem. Todos os Palavra-HMMs constituídos dessa forma são percorridos pelo algoritmo de busca em paralelo. Como o espaço de estados é grande, mesmo para aplicações com vocabulário de tamanho médio, a heurística da busca em feixe é normalmente aplicada para limitar a busca, através da poda (*pruning*) de estados menos prováveis. A combinação do algoritmo de busca e do método de poda utilizados é referida como busca em feixe de Viterbi (*Viterbi beam search*). A decodificação de Viterbi é uma busca síncrona no tempo que processa a fala, segmento a segmento, atualizando todos os estados associados a um segmento antes de passar para o próximo.

Por ser síncrona, a busca em feixe é possivelmente a técnica mais utilizada em sistemas de reconhecimento de voz, desde as referências [36], [37], [38] até hoje em dia. É um algoritmo de busca ao estilo da busca em largura (*breadth-first*), no qual os nós de uma determinada altura h (distância entre um nó e o nó raiz) são analisados antes de passar para nós em uma altura $h+1$ em relação à raiz. Porém, diferente da busca em largura, a busca em feixe expande, a cada passo, apenas os nós que apresentam uma probabilidade alta de sucederem. Apenas estes nós expandidos permanecem no feixe e o restante é ignorado

(podado) aumentando assim a eficiência da busca. A Fig. 2.7, ilustra os passos de busca em feixe. Os nós em cinza são aqueles que foram mantidos na busca. Os nós pontilhados são aqueles que foram explorados, porém pela sua baixa probabilidade foram eliminados.

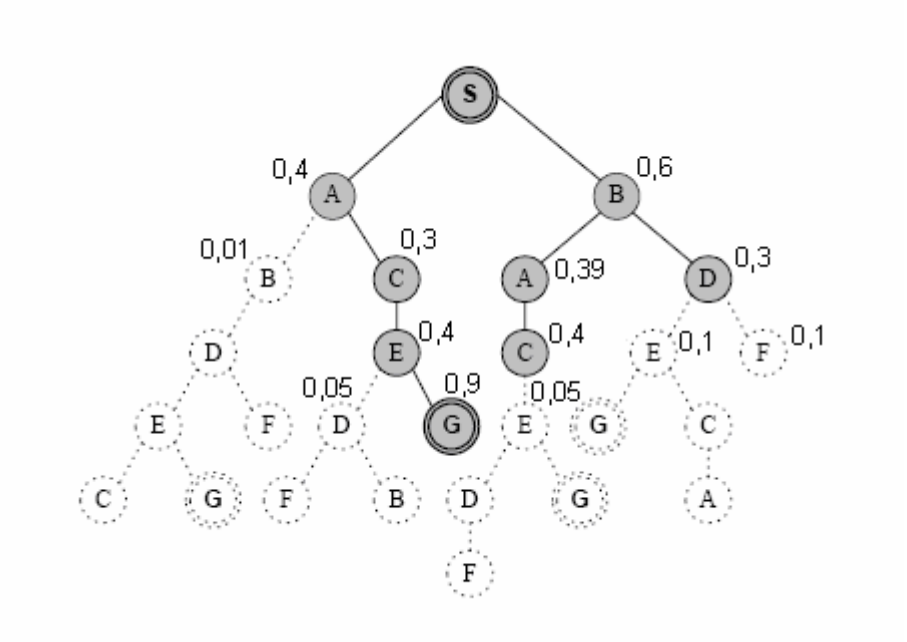

Figura 2.7 – Dinâmica da Busca em Feixe [18]

2.3. Conclusão

Neste capítulo foi feita a apresentação dos sistemas de reconhecimento de voz distribuídos, abordando a extração de atributos e o reconhecimento de voz contínua.

No capítulo seguinte, será feita a apresentação dos codificadores de voz que serão utilizados na montagem dos sistemas de reconhecimento distribuído no ambiente celular e de voz sobre IP.