Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov

Carlos Alberto Ynoguti
Orientador: Prof. Dr. Fábio Violaro

Banca Examinadora:
Prof. Dr. Fábio Violaro – FEEC - UNICAMP
Prof. Dr. Abrahan Alcaim – CETUC – PUC – RIO
Prof Dr. Ivandro Sanches – POLI – USP
Prof. Dr. Luís Geraldo Meloni – FEEC – UNICAMP
Prof. Dr. Lee Luan Ling – FEEC – UNICAMP

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

Campinas, maio de 1999

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Ynoguti, Carlos Alberto
Y69r Reconhecimento de fala contínua usando modelos ocultos de Markov. / Carlos Alberto Ynoguti.-- Campinas, SP: [s.n.], 1999.

Orientador: Fábio Violaro.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Markov, Processos de. 2. Reconhecimento automático da voz. 3. Processamento de sinais – Técnicas digitais. I. Violaro, Fábio. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Resumo

Nos sistemas que constituem o estado da arte na área de reconhecimento de fala predominam os modelos estatísticos, notadamente aqueles baseados em Modelos Ocultos de Markov (*Hidden Markov Models*, HMM). Os HMM's são estruturas poderosas pois são capazes de modelar ao mesmo tempo as variabilidades acústicas e temporais do sinal de voz.

Métodos estatísticos são extremamente vorazes quando se trata de dados de treinamento. Deste modo, nos sistemas de reconhecimento de fala contínua e vocabulário extenso, as palavras são geralmente modeladas a partir da concatenação de sub-unidades fonéticas, pois o número destas é bem menor do que o de palavras, e em uma locução geralmente existem vários exemplos de sub-unidades fonéticas.

O reconhecimento de fala contínua difere do de palavras isoladas, pois neste o locutor não precisa fazer pausas entre as palavras. Deste modo, a determinação das fronteiras entre as palavras e do número destas na locução deve ser feita pelo sistema de reconhecimento. Para isto são utilizados os algoritmos de busca, que podem ter ainda modelos de duração e de linguagem incorporados.

O objetivo deste trabalho é estudar o problema de reconhecimento de fala contínua, com independência de locutor e vocabulário médio (aproximadamente 700 palavras) utilizando HMM's. É investigada a influência de alguns conjuntos de sub-unidades fonéticas, e dos modelos de duração e de linguagem no desempenho do sistema. Também são propostos alguns métodos de redução do tempo de processamento para os algoritmos de busca.

Para a avaliação do sistema foi confeccionada uma base de dados formada de 200 frases foneticamente balanceadas, com gravações de 40 locutores adultos, sendo 20 de cada sexo

Palavras chave: Modelos Ocultos de Markov, reconhecimento de fala contínua, processamento digital de sinais.

Abstract

In the field of continuous speech recognition, current state of art systems make use of statistical methods, mainly those based on Hidden Markov Models (HMM). HMM are powerful due to their ability to model both the acoustic and temporal features of speech signals.

Statistical methods require lots of training samples. For this reason, large vocabulary, continuous speech recognition systems use word models composed by concatenating subunit models. In this approach there are much fewer subunits than words, and many samples of them in a single utterance.

The main difference between continuous speech recognition and isolated words speech recognition is basically in the way that users interact with the system. In isolated words speech recognition, the user needs to make short pauses between words, which is not required for continuous speech recognition systems. The determination of word boundaries, and consequently the number of words in the utterance, take a part of the recognition process in continuous speech recognition systems. For this task searching algorithms are used, and they can also incorporate word duration and language models.

The purpose of this work is to study the problem of speaker independent, medium-size vocabulary (about 700 words), continuous speech recognition using HMM's. The influence of some different subunit sets, word duration model and language model in the overall system performance are investigated. We also propose some methods to alleviate the computational burden in the searching procedure.

To perform system evaluation a multispeaker database (20 male and 20 female) composed of 200 phonetically balanced sentences was created.

Keywords: Hidden Markov Models, continuous speech recognition, digital signal processing.

A meus pais Mituyosi (in memoriam) e Clara e a meus irmãos Sérgio e Cristiane.

Agradecimentos

Ao Prof. Dr. Fábio Violaro pela acolhida e apoio durante os primeiros tempos em uma nova cidade, pela orientação do trabalho, e pelas inúmeras discussões e idéias.

Aos Profs. Drs. José Carlos Pereira e Marcelo Basílio Joaquim pelo apoio e grande ajuda.

À Adriana por seu carinho, paciência e compreensão nos dias difíceis.

Aos colegas do LPDF, Henrique, Fernando, Cairo, Edmilson, Fabrício, Antônio Marcos, Raquel, Irene, Flávio, e Léo pela grande ajuda e por proporcionarem um ambiente de trabalho alegre e descontraído.

Aos colegas e amigos Marcelo, Ricardo, Fábio, Alexandre e Richard pelo apoio e compreensão.

Aos professores e funcionários da FEEC.

Às pessoas que emprestaram suas vozes na confecção da base de dados.

Ao CNPq, pela concessão da bolsa, ao FAEP da UNICAMP pela prorrogação de bolsa concedida, e à FAPESP (processo 97/02740-7) pelo auxílio à pesquisa.

Índice

LISTA DE FIGURAS

LISTA DE TABELAS

1. Introdução.

As interfaces via voz estão rapidamente se tornando uma necessidade. Em um futuro próximo, sistemas interativos irão fornecer fácil acesso a milhares de informações e serviços que irão afetar de forma profunda a vida cotidiana das pessoas. Hoje em dia, tais sistemas estão limitados a pessoas que tenham acesso aos computadores, uma parte relativamente pequena da população, mesmo nos países mais desenvolvidos. São necessários avanços na tecnologia de linguagem humana para que o cidadão médio possa acessar estes sistemas, usando habilidades de comunicação naturais e empregando aparelhos domésticos, tais como o telefone.

Sem avanços fundamentais em interfaces voltadas ao usuário, uma larga fração da sociedade será impedida de participar da era da informação, resultando em uma maior extratificação da sociedade, agravando ainda mais o panorama social dos dias de hoje. Uma interface via voz, na linguagem do usuário, seria ideal pois é a mais natural, flexível, eficiente, e econômica forma de comunicação humana.

Depois de vários anos de pesquisa, a tecnologia de reconhecimento de fala está passando o limiar da praticabilidade. A última década testemunhou um progresso assombroso na tecnologia de reconhecimento de fala, no sentido de que estão se tornando disponíveis algoritmos e sistemas de alto desempenho. Em muitos casos, a transição de protótipos de laboratório para sistemas comerciais já se iniciou.

1.1. Aplicações.

Algumas das principais áreas de aplicação comercial para os sistemas de reconhecimento automático de fala são: ditado, interfaces para computadores pessoais, serviços de telefonia automáticos e aplicações industriais especiais [42]. A principal razão para o sucesso comercial tem sido o aumento na produtividade proporcionado por estes sistemas que auxiliam ou substituem operadores humanos.

1.1.1. Sistemas de ditado de vocabulário extenso.

Os sistemas de ditado de vocabulário extenso podem ser de dois tipos: ditado irrestrito (por exemplo cartas de negócios ou artigos de jornais) e geração de documentos estruturados (por exemplo, receitas médicas, apólices de seguro, relatórios radiológicos, etc). Tais sistemas podem ser dependentes do locutor ou adaptativos desde que se espera que geralmente um único usuário irá utilizá-lo por um período extenso de tempo.

Até bem pouco tempo atrás, os sistemas de palavras isoladas predominaram no mercado. Agora, sistemas de reconhecimento de fala contínua começam a aparecer. Os vocabulários são de aproximadamente 60000 palavras. Estes sistemas são projetados para operar em condições favoráveis (por exemplo, em escritórios, com microfones fixos na cabeça do operador e com cancelamento de ruído).

Para aumentar a taxa de acertos, os sistemas de ditado irrestrito contam com modelos de linguagem estatísticos para favorecer palavras ou sequências de palavras mais frequentes. Os sistemas de domínio específico podem aumentar o seu desempenho incorporando um padrão de documento estruturado para gerar um relatório completo, embora muitas vezes isto exija um processo de planejamento bastante laborioso.

Um sistema de ditado torna-se mais poderoso se possui a habilidade de se adaptar à voz de um determinado usuário (adaptação ao locutor), vocabulário (aprendizado de novas palavras), e tarefas (adaptação do modelo de linguagem).

1.1.2. Interface para computadores pessoais.

A fala tende a se tornar uma componente importante na interface com os computadores. Algumas das possíveis aplicações poderiam ser:

- Fala como atalho: ao invés de abrir um arquivo através de vários níveis de hierarquia, o usuário apenas diz “Abra o estoque”.
- Recuperação de informação: interfaces gráficas são inconvenientes para especificar recuperação de informações baseada em restrições (“encontre todos os documentos de Fábio criados depois de março”)
- Computadores de bolso: à medida em que o tamanho dos computadores diminui (hoje existem palm-tops minúsculos), teclados e mouses tornam-se cada vez mais difíceis de usar, tornando a fala uma alternativa bastante atraente.

Embora o reconhecimento de fala em computadores seja uma alternativa bastante atraente, as interfaces atuais, teclado e mouse, representam uma alternativa madura e extremamente eficiente. É improvável que a fala possa substituir completamente estes dispositivos. Ao invés disso, a nova interface deve combinar estes dispositivos e permitir que o usuário defina qual combinação de dispositivos é a mais adequada para determinada tarefa.

O uso apropriado da fala nos computadores pessoais irá provavelmente requerer o desenvolvimento de um novo conceito de interação com o usuário ao invés de simplesmente modificar as interfaces gráficas existentes.

Uma questão social também está envolvida neste tipo de interface: a dos deficientes físicos. Com interfaces via voz, pessoas que não têm acesso ao computador por causa de suas deficiências, poderiam utilizá-lo normalmente, permitindo um ingresso ao mercado de trabalho e uma competição em pé de igualdade com as outras pessoas.

1.1.3. Sistemas baseados na rede telefônica.

O reconhecimento de fala baseado na rede telefônica oferece um potencial enorme por ser um meio de comunicação extremamente difundido. É também a área tecnicamente mais difícil para o reconhecimento devido à impossibilidade de controle sobre as condições de uso.

Os problemas envolvem uma grande e imprevisível população de usuários, diferenças nos microfones dos aparelhos, a presença de ruído de canal e banda estreita.

Os sistemas mais bem sucedidos são os que se limitam a vocabulários extremamente pequenos, da ordem de 10 a 20 palavras. Para que um sistema seja útil não é necessário um vocabulário muito grande; alguns sistemas tem um vocabulário de apenas duas palavras (“sim” e “não”).

Além do pouco controle sobre a qualidade do sinal, o reconhecimento através da linha telefônica apresenta problemas devido à expectativa dos usuários que o sistema se comporte como um interlocutor humano. Dois exemplos clássicos seriam:

- usuário fala enquanto o sistema ainda está formulando as questões (intromissão), de modo que na hora em que o sistema entra em modo de gravação para coletar a resposta, o usuário já está no meio da resposta ou já terminou de falar
- usuário adiciona palavras à resposta, que não estão no vocabulário do sistema (“sim, por favor”). Neste caso podem ser usadas técnicas de identificação de palavras para conseguir taxas de reconhecimento aceitáveis .

Estes serviços de operação envolvem vocabulários pequenos, diálogo interativo e avisos. As possíveis aplicações seriam: validação de cartões de crédito, compras por catálogo, reservas para hotéis, restaurantes, teatros, passagens aéreas, consultas a telefones e endereços, etc.

1.1.4. Aplicações industriais e sistemas integrados.

Os sistemas de reconhecimento de fala também podem ser utilizados em aplicações mais simples de vocabulário restrito, como o controle de máquinas e dispositivos, abertura e fechamento de portas e válvulas, acendimento de luzes, operações financeiras e outros.

Para muitas aplicações o reconhecimento dependente de locutor é suficiente, desde que um dispositivo particular será utilizado por uma única pessoa durante um período de tempo relativamente extenso, por exemplo um turno de trabalho. Por outro lado, seria conveniente para algumas aplicações que o sistema pudesse fazer reconhecimento de palavras conectadas, uma vez que uma entrada por palavras isoladas pode ser muito lenta e desconfortável.

Dispositivos de reconhecimento de fala podem ser também utilizados como parte de simuladores, permitindo que um sistema automático substitua um treinador humano. Outra aplicação possível é a de sistemas de inspeção móvel e controle de inventário, por exemplo no caso de atividades envolvendo microscopia e trabalho em quartos escuros de fotografia. A cada dia é mais comum ver aparelhos de telefonia celular com discagem por voz (“Ligue-me com o Fábio”).

Estes exemplos significam uma nova era na interação homem-máquina, onde cada vez mais a tecnologia procura criar interfaces que sejam mais naturais ao homem. Com o amadurecimento da tecnologia de reconhecimento de fala, será possível fazer com que todos estes serviços sejam oferecidos de forma segura e eficiente.

1.2. Objetivos e contribuições do Trabalho.

Dentre as várias aplicações citadas para os sistemas de reconhecimento de fala, este trabalho focalizou o problema de reconhecimento de fala contínua, com independência de locutor e vocabulário médio, sendo um caso típico o de editor de texto comandado por voz.

Além do desenvolvimento de um sistema completo para treinamento e reconhecimento, foram estudadas todas as etapas envolvidas no processo, desde o planejamento, gravação e transcrição fonética da base de dados utilizada até a implementação final do sistema.

Também houve a preocupação de se criar um sistema que pudesse ser utilizado por outros pesquisadores, tendo uma interface visual bastante intuitiva e documentação bastante cuidadosa, com o intuito de diminuir o tempo de desenvolvimento e facilitar as pesquisas futuras.

Como contribuições principais deste trabalho pode-se citar a proposta de um conjunto de fones dependentes de contexto consistente e razoavelmente menor do que os trifones propriamente ditos, e a verificação da influência da transcrição fonética das locuções de treinamento no desempenho do sistema. O estudo de todas as etapas do desenvolvimento de um sistema de reconhecimento também proporcionou uma visão bastante ampla e clara dos problemas envolvidos, e serviu para um melhor direcionamento das linhas de pesquisa.

1.3. Conteúdo da Tese.

A tese está organizada da seguinte maneira. No Capítulo 2 é feito um levantamento dos principais problemas observados na tarefa de reconhecimento de fala, com ênfase especial no problema de reconhecimento de fala contínua; é também apresentada uma visão geral do estado da arte atual para os sistemas de reconhecimento

de fala em várias aplicações. O Capítulo 3 discute a questão das bases de dados: como são feitas, como deveriam ser feitas, as dificuldades de confecção, e finalmente os trabalhos realizados para a confecção da base de dados utilizada neste trabalho. No Capítulo 4 é apresentada a teoria sobre modelos ocultos de Markov. O Capítulo 5 trata dos algoritmos de busca com ênfase para o *Level Building* e o *One Step*. O sistema desenvolvido neste trabalho é descrito no Capítulo 6, e os testes e resultados obtidos são apresentados no Capítulo 7. Finalmente, no Capítulo 8 são feitas as análises sobre os resultados e tiradas conclusões a partir destas. Também são feitas sugestões para a continuação das pesquisas a partir das deficiências observadas.

2. O problema do reconhecimento de fala.

O reconhecimento de fala consiste em mapear um sinal acústico, capturado por um transdutor (usualmente um microfone ou um telefone) em um conjunto de palavras.

Os sistemas de reconhecimento de fala podem ser caracterizados por vários parâmetros sendo que alguns dos mais importantes se encontram resumidos na Tabela 1[13].

Tabela 1: Parâmetros típicos usados para caracterizar a capacidade de sistemas de reconhecimento de fala.

Parâmetros	Faixa
Modo de Pronúncia	De palavras isoladas a fala contínua
Estilo de pronúncia	De leitura a fala espontânea
Treinamento	De dependente de locutor a independente de locutor
Vocabulário	De pequeno (< 20 palavras) a grande (> 20000 palavras)
Modelo de linguagem	De estados finitos a sensível a contexto
Perplexidade	De pequena (< 10) a grande (> 100)
SNR	De alta (> 30 dB) a baixa (< 10 dB)
Transdutor	De microfone com cancelamento de ruído a telefone

Um sistema de reconhecimento de palavras isoladas requer que o locutor efetue uma pequena pausa entre as palavras, enquanto que um sistema de reconhecimento de fala contínua não apresenta este inconveniente.

A fala quando gerada de modo espontâneo é mais relaxada, contém mais coarticulações, e portanto é muito mais difícil de reconhecer do que quando gerada através de leitura.

Os sistemas dependentes de locutor necessitam de uma fase de treinamento para cada usuário antes de serem utilizados, o que não acontece com sistemas independentes do locutor, desde que estes já foram previamente treinados com vários locutores.

O reconhecimento torna-se mais difícil à medida em que o vocabulário cresce, ou apresenta palavras parecidas.

Quando a fala é produzida em sequências de palavras, são usados modelos de linguagem para restringir as possibilidades de sequências de palavras. O modelo mais simples pode ser definido como uma máquina de estados finita, onde são explicitadas as palavras que podem seguir uma dada palavra. Os modelos de linguagem mais gerais, que aproximam-se da linguagem natural, são definidos em termos de gramáticas sensíveis a contexto.

Uma medida popular da dificuldade da tarefa, que combina o tamanho do vocabulário e o modelo de linguagem, é a *perplexidade*, grosseiramente definida como a média do número de palavras que pode seguir uma palavra depois que o modelo de linguagem foi aplicado.

Existem também parâmetros externos que podem afetar o desempenho de um sistema de reconhecimento de fala, incluindo as características do ruído ambiente e o tipo e posição do microfone.

O reconhecimento de fala é um problema difícil devido às várias fontes de variabilidade associadas ao sinal de voz [13]:

- *variabilidades fonéticas* : as realizações acústicas dos fonemas, a menor unidade sonora das quais as palavras são compostas, são altamente dependentes do contexto em que aparecem [1]. Por exemplo o fonema /*t*/ em *tatu* tem uma articulação puramente oclusiva, e em *tia*, dependendo do locutor, pode ter uma articulação africada, onde à oclusão se segue um ruído fricativo semelhante ao do início da palavra “chuva”. Além disso, nas fronteiras entre palavras, as variações contextuais podem tornar-se bem mais acentuadas fazendo, por exemplo, com que a frase ‘*a justiça é ...*’ seja pronunciada como ‘*ajusticé...*’

- *variabilidades acústicas*: podem resultar de mudanças no ambiente assim como da posição e características do transdutor.
- *variabilidades intra-locutor*: podem resultar de mudanças do estado físico/emocional dos locutores, velocidade de pronúncia ou qualidade de voz.
- *variabilidades entre-locutores*: originam-se das diferenças na condição sócio - cultural, dialeto, tamanho e forma do trato vocal para cada uma das pessoas.

Os sistemas de reconhecimento tentam modelar as fontes de variabilidade descritas acima de várias maneiras:

- Em termos fonético acústicos, a variabilidade dos locutores é tipicamente modelada usando técnicas estatísticas aplicadas a grandes quantidades de dados de treinamento. Também têm sido desenvolvidos algoritmos de adaptação ao locutor que adaptam modelos acústicos independentes do locutor para os do locutor corrente durante o uso [47][55].
- As variações acústicas são tratadas com o uso de adaptação dinâmica de parâmetros [47], uso de múltiplos microfones [48] e processamento de sinal [13].
- Na parametrização dos sinais, os pesquisadores desenvolveram representações que enfatizam características independentes do locutor, e desprezam características dependentes do locutor [14][18].
- Os efeitos do contexto linguístico em termos fonético-acústicos são tipicamente resolvidos treinando modelos fonéticos separados para fonemas em diferentes contextos; isto é chamado de modelamento acústico dependente de contexto [30].
- O problema da diferença de pronúncias das palavras pode ser tratado permitindo pronúncias alternativas de palavras em representações conhecidas como redes de pronúncia. As pronúncias alternativas mais comuns de cada palavra, assim como os efeitos de dialeto e sotaque são tratados ao se permitir aos algoritmos de busca encontrarem caminhos alternativos de fonemas através destas redes. Modelos

estatísticos de linguagem, baseados na estimativa de ocorrência de sequências de palavras, são geralmente utilizados para guiar a busca através da sequência de palavras mais provável [13].

Outro problema encontrado na tarefa de reconhecimento de fala contínua é o procedimento de decodificação da locução. Este, em sistemas de reconhecimento de fala contínua com vocabulário extenso, tem um custo computacional elevadíssimo, fazendo com que seja necessário buscar maneiras inteligentes de guiar o processo de busca. Este tópico será abordado com mais detalhes na seção seguinte.

2.1. Arquiteturas para reconhecimento de fala.

Atualmente, os algoritmos mais populares na área de reconhecimento de fala baseiam-se em métodos estatísticos. Dentre estes, dois métodos têm se destacado: as redes neurais artificiais (*Artificial Neural Networks*, ANN) [49][54] e os modelos ocultos de Markov (*Hidden Markov Models*, HMM) [5][3][29][40]. Mais recentemente, implementações híbridas que tentam utilizar as características mais favoráveis de cada um destes métodos também têm obtido bons resultados [45].

2.2. Unidades fundamentais.

Em sistemas de vocabulário pequeno (algumas dezenas de palavras), é comum utilizar-se as palavras como unidades fundamentais. Para um treinamento adequado destes sistemas, deve-se ter um grande número de exemplos de cada palavra. Entretanto, para sistemas com vocabulários maiores, a disponibilidade de um grande número de exemplos de cada palavra torna-se inviável. A utilização de sub-unidades fonéticas, tais como fonemas, sílabas, demissílabas, etc, é uma alternativa bastante razoável, pois agora

é necessário ter vários exemplos de cada sub-unidade, e não vários exemplos de cada palavra.

Dois critérios bastante importantes para uma boa escolha de sub-unidades são [30]:

- *consistência*: exemplos diferentes de uma unidade devem ter características similares.
- *treinabilidade*: devem existir exemplos de treinamento suficientes de cada sub-unidade para criar um modelo robusto.

Sub-unidades maiores tais como sílabas, demisssílabas, difones, etc, são consistentes, mas difíceis de treinar, enquanto que unidades menores, tais como os fones, são treináveis, mas inconsistentes.

Uma alternativa que mostrou ser bastante atrativa é a de fones dependentes de contexto [46]. Estas unidades são bastante consistentes, pois levam em consideração o efeito de coarticulação com os fones vizinhos.

Os fones dependentes de contexto, como o nome sugere, modelam o fone em seu contexto. Um contexto geralmente refere-se ao fones imediatamente vizinhos à direita e à esquerda. Um fone dependente do contexto à esquerda é aquele modificado pelo fone imediatamente anterior, enquanto que um fone dependente do contexto à direita é aquele modificado pelo fone imediatamente posterior.

O modelo trifone leva em consideração tanto o contexto à esquerda como o contexto à direita; deste modo, se dois fones têm a mesma identidade mas contextos à esquerda e/ou à direita diferentes, então são considerados trifones distintos.

Estes modelos são em geral insuficientemente treinados devido à sua grande quantidade. Entretanto, como os modelos de trifones são modelos de fones específicos, podem ser interpolados com modelos de fones independentes de contexto, fones dependentes de contexto à esquerda, e fones dependentes de contexto à direita, que são modelos menos consistentes, mas melhor treinados.

2.3. Modelos Ocultos de Markov (HMM's).

A história dos HMM's precede seu uso no processamento de voz e somente mais tarde, gradualmente, foi se tornando bem conhecido e usado no campo da fala. A introdução dos HMM's no campo da voz é usualmente creditada aos trabalhos independentes de Baker na Carnegie Mellon University [5] e Jelinek e colegas na IBM [26].

Os HMM's podem ser classificados em modelos discretos, contínuos e semi-contínuos, de acordo com a natureza dos elementos na matriz de emissão de símbolos (*b*), que são funções de distribuição [41].

Nos modelos discretos, as distribuições são definidas em espaços finitos. Neste caso, as observações são vetores de símbolos de um alfabeto finito de N elementos distintos.

Outra possibilidade é definir distribuições como densidades de probabilidade em espaços de observação contínuos (modelos contínuos). Neste caso, devem ser impostas fortes restrições à forma funcional das distribuições, de modo a se obter um número razoável de parâmetros a serem estimados. A estratégia mais popular é caracterizar as transições do modelo através de misturas de densidades que tenham uma forma paramétrica simples (por exemplo Gaussianas ou Laplacianas), e que possam ser caracterizadas pelo vetor média e pela matriz de covariância. De modo a modelar distribuições complexas desta maneira pode ser necessário usar um grande número destas funções em cada mistura. Isto pode requerer um conjunto de treinamento muito grande para uma estimação robusta dos parâmetros das distribuições.

Nos modelos semicontínuos, todas as misturas são expressas em termos de um conjunto comum de densidades base. As diferentes misturas são caracterizadas somente através de fatores de ponderação diferentes.

2.4. Modelo de duração de palavras.

A idéia de se utilizar um modelo de duração de palavras é penalizar hipóteses levantadas pelo decodificador que estejam fora da duração média (em segundos, por exemplo) da realização de uma dada palavra [40]. Por exemplo, se o decodificador reconheceu a palavra "casa" e atribuiu a ela uma duração de 20 segundos, obviamente esta hipótese deve ser severamente punida, pois está muito longe da realidade.

Para isto, devemos conhecer a priori a duração média de cada uma das palavras que constituem o vocabulário do sistema de reconhecimento. Em sistemas dependentes do locutor, esta estimativa pode ser razoavelmente precisa, mas para sistemas independentes de locutor, torna-se um problema sério estimar a duração média de cada palavra. Além disso, para sistemas com vocabulário grande, a determinação da duração média de cada uma das palavras pode se tornar inviável.

2.5. Algoritmos de decodificação.

A decodificação é um processo de busca no qual uma sequência de vetores correspondentes a características acústicas 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 são comparados com uma sequência de vetores acústicos.

Os algoritmos mais utilizados nesta fase do reconhecimento são todos baseados no algoritmo de Viterbi e, dentre eles, podemos citar: *Level Building* [35], *One Step* [36], *Stack Decoding* [24], entre outros.

Estes modelos crescem com o vocabulário, e podem gerar espaços de busca extremamente grandes, o que torna o processo de busca bastante oneroso em termos computacionais, e portanto lento.

Algumas estratégias para diminuir o custo computacional nesta etapa envolvem procedimentos de poda, tais como o Viterbi *Beam Search* [41].

Deve-se acrescentar que esta etapa do reconhecimento é responsável por praticamente 100% do esforço computacional de um sistema de reconhecimento de fala contínua e, portanto, é a que determina a velocidade final de reconhecimento.

2.6. Modelos de linguagem.

Um sistema de reconhecimento de fala converte o sinal acústico observado em sua representação ortográfica correspondente. O sistema faz a sua escolha a partir de um vocabulário finito de palavras que podem ser reconhecidas. Por simplicidade, assume-se que uma palavra é identificada somente por sua pronúncia [1].

Foi conseguido um progresso dramático na resolução do problema de reconhecimento de fala através do uso de um modelo estatístico da distribuição conjunta $p(W,O)$ da sequência W de palavras pronunciadas e da sequência de informação acústica observada O. Este método é chamado de modelo de fonte-canal. Neste método, o sistema determina uma estimativa $\hat{W}$ da identidade da sequência de palavras pronunciadas a partir da evidência acústica observada O usando a distribuição a posteriori $p(W|O)$. Para minimizar a taxa de erro, o sistema escolhe a sequência de palavras que maximiza a distribuição a posteriori:

[1] Por exemplo, a palavra ‘macaco’ é considerada uma palavra só, embora possa ter mais de um significado (animal ou objeto).

$$\hat{W} = \arg\max_{W} p(W|O) = \arg\max_{W} \frac{p(W)p(O|W)}{p(O)} \quad (1)$$

onde $p(W)$ é a probabilidade da sequência de *n* palavras W e $p(O|W)$ é a probabilidade de observar a evidência acústica O quando a sequência W é pronunciada. A distribuição a priori $p(W)$ de quais palavras poderiam ter sido pronunciadas (a fonte) refere-se ao modelo de linguagem. O modelo da probabilidade de observação $p(O|W)$ (o canal) é chamado de modelo acústico.

2.6.1. Modelos de linguagem *n*-gram.

Para uma dada sequência de palavras $W = \{w_1, ..., w_n\}$ de *n* palavras, pode-se reescrever a probabilidade do modelo de linguagem como:

$$P(W) = P(w_1, ..., w_n) = \prod_{i=1}^{n} P(w_i \mid w_0, ..., w_{i-1}) \quad (2)$$

onde w_0 é escolhido de forma conveniente para lidar com a condição inicial. A probabilidade da próxima palavra w_i depende da história $h_i = (w_1, w_2, ..., w_{i-1})$ das palavras que já foram pronunciadas. Com esta fatoração, a complexidade do modelo de linguagem cresce exponencialmente com o comprimento da história. De modo a obter um modelo mais prático e parcimonioso, a história de palavras pronunciadas é truncada, de modo que apenas alguns termos são utilizados para calcular a probabilidade da próxima palavra seguir a palavra atual.

Os modelo mais bem sucedidos das últimas duas décadas são os modelos *n-gram*, onde somente as *n* palavras mais recentes da história são usadas para condicionar a probabilidade da próxima palavra. O desenvolvimento a seguir refere-se ao caso

particular de gramáticas bigrama (n = 2), A probabilidade de uma sequência de palavras torna-se:

$$P(W) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \tag{3}$$

Para estimar as probabilidades bigrama, pode-se usar um grande corpus de textos para estimar as respectivas frequências bigrama:

$$f_2(w_2 \mid w_1) = \frac{c_{12}}{c_1} \tag{4}$$

onde c_{12} é o número de vezes que a sequência de palavras $\{w_1, w_2\}$ é observada e c_1 é o número de vezes que w_1 é observada. Para um vocabulário de V palavras existem V^2 bigramas possíveis, o que para um vocabulário de 20000 palavras significa 400 milhões de bigramas. Muitos destes bigramas não serão observados no corpus de treinamento, e deste modo estes bigramas não observados irão ter probabilidade zero quando se usa a frequência bigrama como uma estimativa da probabilidade bigrama. Para resolver este problema, é necessário uma estimativa suavizada da probabilidade de eventos não observados. Isto pode ser feito pela interpolação linear das frequências bigrama e unigram e uma distribuição uniforme no vocabulário.

$$p(w_2 \mid w_1) = \lambda_2 f_2(w_2 \mid w_1) + \lambda_1 f_1(w_2) + \lambda_0 \frac{1}{V} \tag{5}$$

onde $f_2(\,)$ e $f_1(\,)$ são estimadas pela razão das contagens bigrama e unigram apropriadas. Os pesos (λ_0, λ_1 e λ_2) da interpolação linear são estimados a partir de dados de validação: maximizando a probabilidade de novos dados diferentes daqueles usados

para estimar as frequências *n-gram*. O algoritmo *forward-backward* pode ser usado para resolver este problema de estimação de máxima verossimilhança.

No trabalho de modelamento de linguagem têm sido usadas bases de dados de um milhão a 500 milhões de palavras, correspondendo a vocabulários de 1000 a 267000 palavras distintas, para construir modelos trigrama [13]. Para gramáticas do tipo bigrama as necessidades são um pouco menores, mas ainda astronômicas.

2.6.2. Perplexidade.

Na comparação de modelos de linguagem, é importante ser capaz de quantificar a dificuldade que estes impõem ao sistema de reconhecimento. Um modo de se fazer isso é utilizá-los em um sistema de reconhecimento e determinar qual deles fornece a menor taxa de erro. Este é ainda a melhor maneira de avaliar um modelo de linguagem, embora seja um método altamente custoso.

Os modelos de linguagem tendem a minorar as incertezas (diminuir a entropia) do conteúdo das sentenças e facilitar o reconhecimento. Por exemplo, se existem, em média, muito poucas palavras que podem seguir uma dada palavra em um modelo de linguagem, o sistema de reconhecimento terá menos opções para verificar, e o desempenho será melhor do que se existissem muitas palavras possíveis. Este exemplo sugere que uma medida conveniente da dificuldade de um modelo de linguagem deva envolver alguma medida do número médio de palavras que possam seguir outras. Se o modelo de linguagem for visto como um grafo, com terminais associados a transições entre palavras, por exemplo, então esta medida estaria relacionada com o fator de ramificação médio em todos os pontos de decisão do grafo. Grosseiramente falando, esta é a quantidade medida pela *perplexidade*, formalizada a seguir.

Um modelo estocástico formal de linguagem gera sequências terminais com certas probabilidades. Estas sequências terminais podem ser vistas como realizações de um processo estocástico estacionário discreto cujas variáveis aleatórias assumem valores discretos. Estes valores discretos correspondem aos terminais individuais, e o tempo

indica simplesmente a posição do terminal aleatório na sequência de palavras. Por simplicidade, vamos assumir que os terminais correspondam a palavras, e este processo aleatório será indicado por $\underline{w}$. Se existem W palavras possíveis, $w_1,...,w_W$, a entropia associada com este processo aleatório ou "fonte" é dada por

$$\begin{aligned} H(\underline{w}) &= -E\{\log_2 P(\underline{w}(\cdot) = w_i)\} \\ &= -\sum_{i=1}^{W} P(\underline{w}(\cdot) = w_i) \log_2 P(\underline{w}(\cdot) = w_i) \end{aligned} \tag{6}$$

onde $\underline{w}(\cdot)$ é uma variável aleatória arbitrária em $\underline{w}$ se a fonte tem variáveis aleatórias independentes e identicamente distribuídas. Se não, a entropia é dada por

$$\begin{aligned} H(\underline{w}) &= -\lim_{N\to\infty} \frac{1}{N} E\left\{\log P\left(\underline{w}_1^N = w_1^N\right)\right\} \\ &= -\lim_{N\to\infty} \frac{1}{N} \sum_{w_1^N} P\left(\underline{w}_1^N = w_1^N\right) \log P\left(\underline{w}_1^N = w_1^N\right) \end{aligned} \tag{7}$$

onde $\underline{w}_1^N$ denota a sequência de variáveis aleatórias $\underline{w}(1),...,\underline{w}(N)$, e w_1^N denota a realização parcial $w(1),...,w(N)$, e a soma é tomada sobre todas estas realizações. Desde que as palavras em um modelo de linguagem não são independentes e nem equiprováveis, usamos (7) ao invés de (6). Para uma fonte ergódica, a entropia pode ser calculada utilizando-se uma média temporal

$$H(\underline{w}) = -\lim_{N\to\infty} \frac{1}{N} \log_2 P\left(\underline{w}_1^N = w_1^N\right) \tag{8}$$

Na prática, quanto mais longa a sentença (N maior) utilizada para estimar H, melhor será a estimativa; H representa o número médio de bits de informação inerente a

uma palavra no modelo de linguagem. Isto significa que, em média, $H(\underline{w})$ bits precisam ser extraídos dos dados acústicos para reconhecer uma palavra.

As probabilidades $P\left(\underline{w}_1^N = w_1^N\right)$ são desconhecidas e precisam ser estimadas a partir de dados de treinamento (que podem ser vitos como exemplos de produções do modelo de linguagem). Chamando as estimativas de $\hat{P}\left(\underline{w}_1^N = w_1^N\right)$, e a medida de entropia resultante de $\hat{H}(\underline{w})$, temos

$$\hat{H}(\underline{w}) = -\lim_{N\to\infty} \frac{1}{N} \log_2 \hat{P}\left(\underline{w}_1^N = w_1^N\right) \tag{9}$$

Pode-se mostrar que $\hat{H} \geq H$ se $\underline{w}$ for um processo ergódico.

Embora a entropia forneça uma medida de dificuldade perfeitamente válida, na área de processamento de fala, prefere-se usar a *perplexidade*, definida como

$$Q(\underline{w}) \stackrel{def}{=} 2^{\hat{H}(\underline{w})} \approx \frac{1}{\sqrt[N]{\hat{P}\left(w_1^N\right)}} \tag{10}$$

para algum N grande. Para verificar o sentido desta medida, note que se o modelo de linguagem tem W palavras equiprováveis que ocorrem independentemente em qualquer sequência de palavras, segue de (6) que a quantidade de entropia em qualquer sequência é dada por

$$H(\underline{w}) = \log_2 W \tag{11}$$

O tamanho do vocabulário neste caso está relacionado com a entropia através da seguinte expressão:

$$W = 2^{H(\underline{w})} \tag{12}$$

Comparando (12) e (10), pode-se ver que a perplexidade de um modelo de linguagem pode ser interpretada como o tamanho do vocabulário (número de terminais) em outro modelo de linguagem com palavras equiprováveis e independentes, que seja igualmente difícil de reconhecer. Portanto a perplexidade indica um fator de ramificação médio de um modelo de linguagem modelado por $\underline{w}$.

A perplexidade de um modelo de linguagem depende do domínio de discurso. Na Tabela 2 tem-se um quadro comparativo para diversas aplicações [13]:

Tabela 2: Perplexidades típicas para vários domínios.

Domínio	Perplexidade
Radiologia	20
Medicina de emergência	60
Jornalismo	105
Fala geral	247

2.7. Estado da arte.

O desempenho dos sistemas de reconhecimento de fala é tipicamente descrito em termos de taxa de erros de palavra *E*, definida como [41]:

$$E = \frac{S + I + D}{N} 100 \tag{13}$$

onde *N* é o número total de palavras no conjunto de teste, *S*, *I* e *D* são o número total de substituições, inserções e deleções, respectivamente.

A última década tem testemunhado um progresso significativo na tecnologia de reconhecimento de fala. As taxas de erro de palavra caem de um fator de 2 a cada dois

anos. Foi feito um progresso substancial na tecnologia básica, o que levou a vencer as barreira de independência de locutor, fala contínua e vocabulários extensos.

Existem vários fatores que contribuíram para este rápido progresso.

- A chegada da era do HMM. O HMM é poderoso no sentido de que, com a disponibilidade de dados de treinamento, os parâmetros do modelo podem ser treinados automaticamente para fornecer um desempenho ótimo.
- Foi feito um grande esforço no sentido de desenvolver grandes bases de dados de fala para o desenvolvimento, treinamento e avaliação de sistemas.
- Estabelecimento de normas de avaliação de desempenho. Até uma década atrás, os pesquisadores treinavam e testavam seus sistemas usando dados coletados localmente, e não foram muito cuidadosos em delinear os conjuntos de treinamento e testes. Consequentemente, era muito difícil comparar o desempenho dos vários sistemas, e ainda, o desempenho de um sistema era geralmente degradado quando este era apresentado a dados novos. A recente disponibilidade de grandes bases de dados no domínio público, associada à especificação de padrões de avaliação, resultou em uma documentação uniforme de resultados de testes, contribuindo para uma maior confiabilidade na monitoração dos progressos alcançados.
- Os avanços na tecnologia dos computadores influenciaram indiretamente o progresso nesta área. A disponibilidade de computadores rápidos com grandes capacidades de memória permitiu aos pesquisadores realizar várias experiências em larga escala e em um curto espaço de tempo. Isto significa que o intervalo de tempo entre uma idéia e a sua implementação e avaliação foi bastante reduzido. De fato, sistemas de reconhecimento de fala com desempenho razoável podem rodar em microcomputadores comuns em tempo real, sem hardware adicional, um fato inimaginável a alguns anos atrás.

Talvez a tarefa mais popular, e potencialmente mais útil, com baixa perplexidade ($PP = 11$) é o reconhecimento de dígitos conectados. Para o inglês americano, o reconhecimento independente de locutor de sequências de dígitos pronunciados de

forma contínua e restringido à largura de banda telefônica pode alcançar uma taxa de erro de 0,3% quando o comprimento da sequência é conhecido.

Uma das tarefas de média perplexidade mais conhecidas é a de 1000 palavras chamada de *Resource Management*, na qual podem-se fazer indagações sobre vários navios no oceano Pacífico. O melhor desempenho independente de locutor nesta tarefa é de menos de 4%, usando um modelo de linguagem de pares de palavras que limita as palavras possíveis que seguem uma dada palavra (*PP* = 60). Mais recentemente, os pesquisadores começaram a estudar a questão do reconhecimento de fala espontânea. Por exemplo, no domínio do Serviço de Informação de Viagens Aéreas (*Air Travel Information Service*, ATIS), foram relatadas taxas de erros de menos de 3% para um vocabulário de aproximadamente 2000 palavras e um modelo de linguagem bigrama com uma perplexidade por volta de 15.

Tarefas com alta perplexidade, com vocabulários de milhares de palavras, são destinadas principalmente para aplicações de ditado. Depois de trabalhar em sistemas de palavras isoladas, dependentes de locutor, por muitos anos, a comunidade tem voltado suas atenções desde 1992 para o reconhecimento de fala contínua para grandes vocabulários (20.000 palavras ou mais), alta perplexidade (*PP* ≈ 200), independente de locutor. O melhor sistema em 1997 conseguiu uma taxa de erro de 9,9% em testes realizados regularmente nos EUA através do Departamento de Defesa. [39].

3.Base de dados.

3.1. Introdução.

A linguagem falada é a forma mais natural de comunicação humana. Sua estrutura é moldada pelas estruturas fonológicas, sintáticas e prosódicas da língua, do ambiente acústico, do contexto em que a fala está sendo produzida (por exemplo, as pessoas falam de maneira diferente em ambientes ruidosos e silenciosos), e do canal através do qual viaja (telefone, microfone, diretamente pelo ar, etc.).

A fala é produzida de maneira diferente por cada pessoa, sendo as variações devidas ao dialeto, forma e tamanho do trato vocal, ritmo de pronúncia, entre outros fatores. Ainda, os padrões de fala são modificados pelo ambiente físico, contexto social, e estado físico e emocional das pessoas.

As tecnologias mais promissoras na área de reconhecimento de fala (redes neurais e HMM's) utilizam métodos de modelagem estatística que aprendem por exemplos, exigindo conjuntos de dados de treinamento extremamente grandes, que cubram todas estas variações.

O efeito causado por variáveis não modeladas ou mal modeladas (tais como diferenças de canal ou microfones, palavras fora do vocabulário, sub-unidades fonéticas mal treinadas) no desempenho dos sistemas de reconhecimento de fala é devastador. Deste modo, para fornecer exemplos em número suficiente para que os métodos estatísticos funcionem adequadamente, a base de dados precisa ser extremamente

grande e, consequentemente, custosa, tanto em termos de trabalho como em termos financeiros.

Estes altos custos só podem ser arcados por um esforço conjunto de empresas, instituições de pesquisa e agências financiadoras, de modo a evitar duplicação de esforços e distribuir as tarefas. Para envolver um número maior de agentes neste processo, é necessário que este material não seja direcionado a um sistema ou tarefa específicos, mas atender as necessidades de vários grupos e linhas de pesquisa e desenvolvimento, em diversas áreas do conhecimento (síntese e reconhecimento de fala, estudos fonéticos, estudos linguísticos, etc.).

Na Europa, o projeto EUROM_1 congregou esforços de 8 países europeus: Itália, Inglaterra, Alemanha, Holanda, Dinamarca, Suécia, Noruega e França, com a adesão posterior de Grécia, Espanha e Portugal. A base de dados foi criada com o mesmo número de locutores (30 homens e 30 mulheres), escolhidos através dos mesmos critérios e gravados em condições acústicas semelhantes, e no mesmo formato.

Ainda, em Portugal, foi criada uma base de dados chamada BD-PUBLICO (Base de Dados em Português eUropeu, vocaBulário Largo, Independente do orador e fala COntínua), com aproximadamente 10 milhões de palavras em aproximadamente 156 mil frases, pronunciadas por 120 locutores (60 de cada sexo). Como não poderia deixar de ser, esta base foi confeccionada através do esforço conjunto de instituições de pesquisa, órgãos governamentais e também empresas do setor privado.

Nos EUA também foi feito um grande esforço neste sentido, e já existem disponíveis no domínio público, várias bases de dados (TIMIT, TI-DIGITS, SWITCHBOARD, etc.) para desenvolvimento e teste de sistemas.

A disponibilidade destas bases impulsionou de forma expressiva o desenvolvimento da tecnologia de fala, não só devido ao fato de os centros de pesquisa não terem que criar suas próprias bases de dados, um trabalho por si só extremamente árduo, caro e demorado, como também pela possibilidade de comparar os resultados de cada nova idéia de uma forma estatisticamente significativa.

No caso do Brasil este tipo de consórcio ainda não foi sequer cogitado, e os pesquisadores têm que desenvolver seus trabalhos como os americanos faziam há 20

anos atrás: com bases caseiras e pequenas, que tentam cobrir os fenômenos mais significativos da língua falada, na maioria dos casos sem sucesso.

3.2. Encaminhamentos futuros.

Os desafios em linguagem falada são muitos. Um desafio básico está na definição da metodologia - como projetar bases de dados compactas que possam ser utilizadas em várias aplicações; como projetar bases de dados que possam ser comparadas em várias línguas; como selecionar locutores para que se tenha uma população representativa em relação a vários fatores, tais como sotaque, dialeto, e modo de pronúncia; como selecionar as frases a serem pronunciadas de modo a cobrir todas as aplicações; como selecionar um conjunto de dados de teste estatisticamente representativo para a avaliação dos sistemas.

Outro desafio é desenvolver padrões para transcrever as locuções em diferentes níveis e entre línguas diferentes: estabelecer conjuntos de símbolos, convenções de alinhamento, definir níveis de transcrição (acústica, fonética, de palavras, e outros), convenções para prosódia e tom, convenções para controle de qualidade das transcrições (por exemplo várias pessoas transcrevendo as mesmas locuções para uma estatística confiável). Também seria interessante classificar as gravações de acordo com o ambiente em que foram feitas, assim como o canal utilizado (ambientes silenciosos ou ruidosos, com música ambiente, gravações feitas através da linha telefônica, etc.).

No caso brasileiro, ainda é necessário juntar os esforços para obter pelo menos uma base de dados padrão, para que os pesquisadores possam comparar métodos e resultados, e assim evitar duplicações de esforços.

3.3. Projeto e confecção da base de dados.

Com dito anteriormente, não se tem disponível para a língua portuguesa uma base de dados de referência sobre a qual se possa desenvolver e testar o desempenho dos sistemas de reconhecimento de fala, tornando-se necessário confeccionar nossas próprias bases de dados.

Por um lado, isto significa um grande dispêndio de tempo e trabalho, que poderiam ser utilizados na elaboração, desenvolvimento e avaliação de novas idéias. Por outro lado, o planejamento e a confecção de uma base de dados traz uma compreensão valiosa da forma com que as pessoas interagem com um sistema de reconhecimento de fala. As variações de pronúncia e qualidade de voz devido à presença de um microfone, condição sócio-cultural, região de origem, estado emocional e até à hora do dia ficam bem claras quando se confecciona uma base de dados relativamente grande.

Os trabalhos de confecção da base de dados consistiram de:

- escolha das frases
- escolha dos locutores
- gravação das locuções
- transcrição fonética

3.3.1. Escolha das frases.

As frases foram escolhidas segundo o trabalho realizado por Alcaim et. al. [1]. Neste, foram criadas 20 listas de 10 frases foneticamente balanceadas, segundo o português falado no Rio de Janeiro, listadas no Apêndice A. Nestas listas, contou-se 694 palavras distintas.

O termo foneticamente balanceado, neste caso, significa que a lista de frases gerada tem uma distribuição fonética similar àquela encontrada na fala espontânea. Esta distribuição foi levantada a partir da transcrição fonética de gravações de inquéritos, obtidas a partir do projeto NURC-RJ (Projeto de Estudo da Norma Linguística Urbana culta na cidade do Rio de Janeiro) [10].

3.3.2. Locutores.

Para a confecção da base de dados, foram selecionados 40 locutores adultos, sendo 20 homens e 20 mulheres. A maioria dos locutores nasceu no interior do estado de São Paulo, embora alguns sejam nativos de outros estados (Pernambuco, Ceará, Paraná e Amazonas). A maioria tem o nível superior, e todos tem pelo menos o segundo grau completo. Um resumo informativo de cada um dos locutores pode ser encontrado no Apêndice B.

Os locutores foram divididos igualmente em 5 grupos, ou seja, 4 homens e 4 mulheres para cada grupo. Para cada grupo foram designadas 4 das 20 listas da base de dados da seguinte forma: as primeiras 4 listas para o primeiro grupo, as 4 seguintes para o segundo grupo, e assim por diante. Desta forma, cada locutor pronunciou no total 40 frases, e cada frase foi repetida por 8 locutores diferentes.

Um locutor extra do sexo masculino completa a base de dados. Este locutor pronunciou todas as 200 frases, repetindo-as 3 vezes. Estas locuções foram utilizadas para testes com dependência de locutor.

3.3.3. Gravações.

As gravações foram realizadas em ambiente relativamente silencioso, com um microfone direcional de boa qualidade, utilizando uma placa de som SoundBlaster AWE

64. A taxa de amostragem utilizada foi de 11,025 kHz, e resolução de 16 bits. Os dados foram armazenados em formato Windows PCM (WAV).

3.3.4. Transcrição Fonética.

A transcrição fonética foi feita manualmente para cada locução, utilizando programa de visualização gráfica do espectrograma e forma de onda do sinal, e fones de ouvido para audição da mesma.

As sub-unidades utilizadas nesta tarefa são mostradas na Tabela 3. É importante frisar que os fones utilizados na transcrição fonética deste trabalho e daquele realizado por Alcaim et al [1] não são os mesmos. No presente trabalho foi utilizado um conjunto menor de sub-unidades fonéticas, resultante da fusão de algumas classes propostas em [1], principalmente entre as vogais.

Mesmo com estas restrições, pode-se observar que, de uma forma geral, o levantamento dos fones a partir da transcrição fonética da base de dados gravada acompanhou a distribuição encontrada em [1]. Entretanto, a comparação da frequência relativa da ocorrência dos fones mostra algumas diferenças significativas, possivelmente decorrentes das variações regionais de pronúncia dos locutores. Na Figura 1, tem-se um histograma comparativo para a ocorrência dos fones em ambos os casos.

Considerando que a maioria dos locutores selecionados para este trabalho tem origem no estado de São Paulo, pode-se considerar que é uma base “paulista”, e como o trabalho do Prof. Alcaim foi realizado somente com locutores cariocas, pode-se considerar que é uma base ‘carioca’. Assim, com ressalvas, pode-se fazer algumas comparações interessantes:

- a diferença de pronúncia do ‘s’ entre consoantes é bem visível se observarmos os histogramas correspondentes aos fones ‘s’ e ‘x’.
- idem para os fones ‘z’ e ‘j ‘
- idem para os fones ‘r’ e ‘rr’.

Tabela 3: sub-unidades acústicas utilizadas na transcrição fonética das locuções, com exemplos e frequências relativas de ocorrência, segundo Alcaim et. al. [1] e aquelas encontradas na transcrição fonética da base de dados coletada. Também são listados os números de ocorrências observados para cada sub-unidade.

Fone	Símbolo utilizado	Exemplo	Frequência Relativa (%)		Número de ocorrências
			Alcaim et. al.	Observada	
a	a	***a*** çafrão	12,94	13,91	6031
e	e	***e*** levador	4,82	2,15	933
ɛ	E	p ***e*** le	1,91	6,35	2785
i	i	s ***i*** no	8,57	1,90	821
j	y	fu ***i***	3,13	0,95	410
o	o	b ***o*** lo	2,71	4,14	1798
ɔ	O	b ***o*** la	1,00	6,23	2691
u	u	l ***u*** a	8,69	2,57	1124
α̃	an	maç ***ã***	2,12	4,04	1773
ẽ	en	s ***en*** ta	2,30	1,16	501
ĩ	in	p ***in*** to	3,23	0,69	296
õ	on	s ***om*** bra	0,75	8,41	3648
ũ	un	um	2,50	1,98	860
b	b	***b*** ela	1,09	1,18	511
d	d	***d*** ádiva	2,64	3,14	1346
dʒ	D	***d*** iferente	1,92	1,49	665
f	f	***f*** eira	1,46	1,44	625
g	g	***g*** orila	0,93	0,87	378
ʒ	j	***j*** iló	1,32	0,75	325
k	k	***c*** achoeira	4,19	3,63	1575
l	l	***l*** eão	1,72	1,91	830
ʎ	L	***lh*** ama	0,21	0,35	152
m	m	***m*** ontanha	4,12	3,77	1637
n	n	***n*** évoa	2,40	2,26	982
ɲ	N	i ***nh*** ame	0,68	0,42	185
p	p	***p*** oente	2,29	2,49	1081
r	r	ce ***r*** a	3,58	4,05	1759
r̄	rr	ce ***rr*** ado	2,06	0,89	363
R	R	ca ***r*** ta	-	1,32	598
s	s	***s*** apo	4,18	6,52	2832
t	t	***t*** empes ***t*** ade	3,94	4,02	1737
tʃ	T	***t*** igela	1,44	1,20	531
v	v	***v*** erão	1,23	1,51	656
ʃ	x	***ch*** ave	2,12	0,32	132
z	z	***z*** abumba	1,81	1,96	859

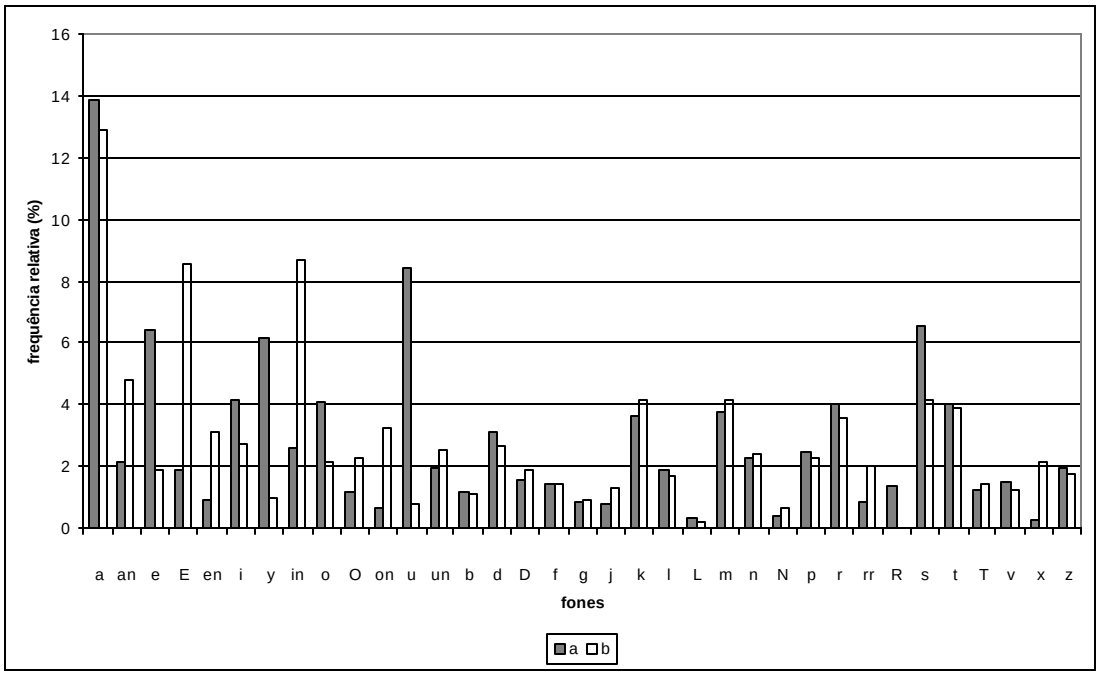


Figura 1: Histograma comparativo da ocorrência de fones nos trabalhos atual a) e os realizados em [1] b).

4. Modelos Ocultos de Markov.

A teoria relativa aos modelos ocultos de Markov já é bem conhecida e extensivamente documentada. Desta forma, neste capítulo são apresentados apenas alguns conceitos básicos e notações importantes para a compreensão das seções posteriores. Textos com explicações bastante claras e precisas podem ser encontrados em [40] e [15].

Em um sistema estatístico de reconhecimento de fala contínua, geralmente as palavras do vocabulário são representadas através de um conjunto de modelos probabilísticos de unidades linguísticas elementares (por exemplo fones). Uma sequência de parâmetros acústicos, extraídos de uma locução, é vista como uma realização de uma concatenação de processos elementares descritos por Modelos Ocultos de Markov (em inglês, *Hidden Markov Models*, HMM). Um HMM é uma composição de dois processos estocásticos, uma cadeia de Markov oculta, relacionada à variação temporal, e um processo observável, relacionado à variabilidade espectral. Esta combinação provou ser poderosa para lidar com as fontes mais importantes de ambiguidade, e flexível o suficiente para permitir a realização de sistemas de reconhecimento com dicionários extremamente grandes (dezenas de milhares de palavras) [13].

4.1. Estrutura de um HMM.

Um HMM é definido como um par de processos estocásticos (**X**,**Y**). O processo **X** é uma cadeia de Markov de primeira ordem, e não é diretamente observável, enquanto que o processo **Y** é uma sequência de variáveis aleatórias que assumem valores no espaço de parâmetros acústicos (*observações*).

Um HMM gera sequências de observações pulando de um estado para outro, emitindo uma observação a cada salto. Em geral, para o reconhecimento de fala, é utilizado um modelo simplificado de HMM conhecido como modelo *left-right*, ou modelo de Bakis [15], no qual a sequência de estados associada ao modelo tem a propriedade de, à medida que o tempo aumenta, o índice do estado aumenta (ou permanece o mesmo), isto é, o sistema caminha da esquerda para a direita no modelo (veja Figura 2)

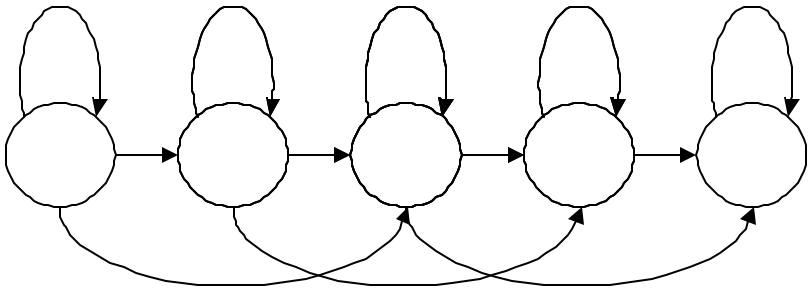


Figura 2: modelo de Bakis para um HMM left-right de 5 estados

São usadas duas formas ligeiramente diferentes para os HMM's. Uma delas usualmente (mas nem sempre) utilizada no processamento acústico (modelamento do sinal) emite uma observação no instante de chegada ao estado. A outra forma , geralmente utilizada em processamento de linguagem, emite uma observação durante a transição. A forma de estado emissor é também chamada de máquina de Moore na teoria de autômatos, enquanto que a forma de transição emissora é uma máquina de Mealy [20]. Neste trabalho, seguindo a tendência geral, foi utilizada a forma de Moore. Na

Figura 3 tem-se um exemplo de cada uma destas formas para um modelo HMM *left-right* de 3 estados.

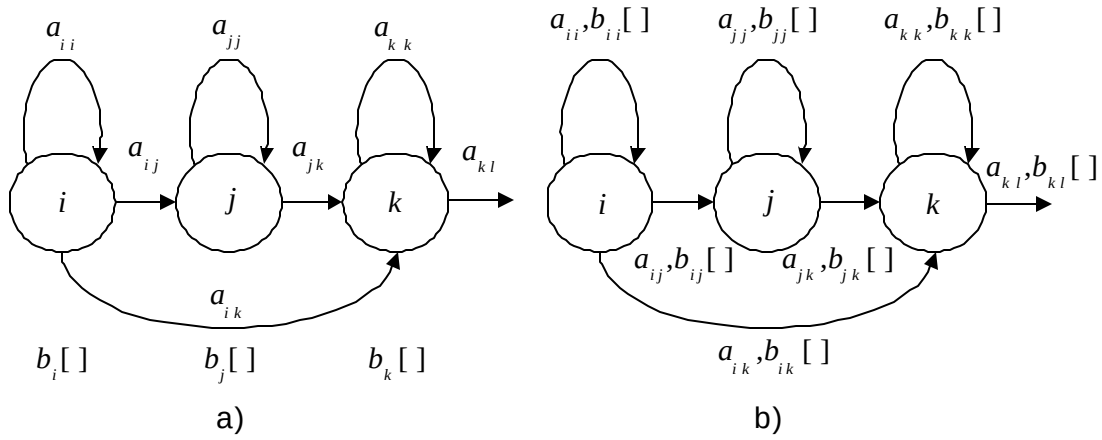


Figura 3: formas de Moore a) e Mealy b) para um HMM com 3 estados.

Na tarefa de reconhecimento de fala, geralmente são adotadas duas simplificações da teoria de modelos de Markov, que podem ser formalizadas da seguinte maneira [15]:

- *Hipótese de Markov de primeira ordem*: a história não tem influência na evolução futura da cadeia se o presente é especificado.
- *Hipótese de independência das saídas*: nem a evolução da cadeia nem as observações passadas influenciam a observação atual se a última transição da cadeia é especificada.

Estas duas hipóteses podem ser escritas da seguinte maneira: seja $y \in \mathbf{Y}$ a variável que representa as observações e $i, j \in \mathbf{X}$ as variáveis que representam os estados do modelo. Então, o modelo pode ser representado pelos seguintes parâmetros:

$$A \equiv \left\{ a_{ij} \mid i, j \in \mathbf{X} \right\} \tag{14}$$

$$B \equiv \{b_i(y) \mid i \in \mathbf{X}, y \in \mathbf{Y}\} \tag{15}$$

$$\Pi \equiv \{\boldsymbol{p}_i \mid i \in \mathbf{X}\} \tag{16}$$

onde A é a matriz com as probabilidades de transição, B é a matriz de densidades de probabilidade de emissão dos símbolos de saída, e Π é a matriz de probabilidades iniciais, com as seguintes definições

$$a_{ij} \equiv P(X_t = j \mid X_{t-1} = i) \tag{17}$$

$$b_j(y) \equiv p(Y_t = y \mid X_t = j) \tag{18}$$

$$\boldsymbol{p}_i \equiv P(X_0 = i) \tag{19}$$

4.2. Tipos de HMM's.

Os HMM's podem ser classificados de acordo com a natureza dos elementos da matriz *B*, que são funções densidade de probabilidade.

Nos HMM's discretos as densidades de probabilidades são definidas em espaços finitos. Neste caso, as observações são vetores de símbolos de um alfabeto finito de N elementos diferentes.

Outra possibilidade é definir as densidades de probabilidade em espaços de observação contínuos. Neste caso é necessário impor severas restrições na forma funcional das densidades de modo a ter um número manipulável de parâmetros estatísticos para estimar. A aproximação mais popular consiste em caracterizar as densidades de emissão do modelo como misturas de densidades base *g* de uma família *G* com uma forma paramétrica simples. As densidades base $g \in G$ são geralmente Gaussianas ou Laplacianas, e podem ser parametrizadas pelo vetor média e pela matriz de covariância. HMM's com este tipo de distribuição são chamados de HMM's contínuos. De modo a modelar distribuições complexas desta maneira é necessário usar um grande número de densidades base em cada mistura. Os problemas que surgem quando o corpus de treinamento não é suficientemente grande podem ser aliviados pelo compartilhamento de distribuições entre emissões de estados diferentes [23].

Nos modelos semicontínuos, todas as misturas são expressas em termos de um conjunto comum *C* de densidades base. Neste caso, as misturas são diferenciadas pelos pesos atribuídos a cada uma das funções base de *C*.

O cálculo das probabilidades com modelos discretos é mais rápido do que com modelos contínuos, embora seja possível acelerar o cálculo das misturas de densidades aplicando a quantização vetorial nas gaussianas das misturas [15].

Levando em consideração o grande apetite por exemplos de treinamento dos modelos contínuos e o fato de a base de dados utilizada ser relativamente pequena, optou-se por utilizar a forma discreta neste trabalho.

4.3. Treinamento dos HMM's.

A estimação dos parâmetros dos HMM's, como em todos os sistemas estocásticos, é baseada em exemplos de treinamento e é geralmente feita utilizando o algoritmo *forward-backward* [40], também conhecido como algoritmo Baum-Welch.

O critério utilizado para a reestimação dos parâmetros é o de máxima verossimilhança ML (*Maximum Likelihood*), que consiste em aumentar, a cada época de

treinamento, a probabilidade a posteriori, ou seja, a probabilidade do modelo gerar a sequência de observações.

4.4. Reconhecimento de fala utilizando HMM's.

Dada uma locução de entrada, um sistema de reconhecimento de fala gera hipóteses de palavras ou sequências de palavras. Destas hipóteses pode resultar uma única sequência de palavras, uma coleção de *n* melhores sequências de palavras, ou uma treliça de hipóteses de palavras parcialmente superpostas. Isto é feito num processo de busca no qual se compara uma sequência de vetores de características acústicas com os modelos das palavras que estão no vocabulário do sistema.

Em geral, o sinal de fala e suas transformações não exibem indicações claras sobre as fronteiras das palavras, de modo que a detecção destas fronteiras faz parte do processo de geração de hipóteses realizado no procedimento de busca. No procedimento de geração de hipóteses, todos os modelos de palavras são comparados com uma sequência de vetores acústicos. Em um sistema probabilístico, a comparação entre uma sequência acústica e um modelo envolve o cálculo da probabilidade que o modelo associa a uma dada sequência. Neste processo, as seguintes quantidades são utilizadas:

- $\alpha_t\left(\mathbf{y}_1^T, i\right)$: probabilidade de observar a sequência de observação parcial $\mathbf{y}_1^t$ [2] e estar no estado *i* no instante *t* (sendo que a sequência de observação total é dada por $\mathbf{y}_1^T$)

$$\alpha_t\left(\mathbf{y}_1^T, i\right) \equiv \begin{cases} P\left(X_0 = i\right), & t = 0 \\ P\left(X_t = i, \mathbf{Y}_1^t = \mathbf{y}_1^t\right), & t > 0 \end{cases} \tag{20}$$

[2] A notação y_h^k refere-se à sequência de vetores acústicos $\left[y_h, y_{h+1}, ..., y_k\right]$. Esta notação será utilizada daqui em diante.

- $\beta_t\left(y_1^T, i\right)$: probabilidade de observar a sequência de observação parcial y_{t+1}^T dado que o modelo está no estado *i* no instante *t*.

$$\beta_t\left(y_1^T, i\right) \equiv \begin{cases} P\left(Y_{t+1}^T = y_{t+1}^T \mid X_t = i\right), & t < T \\ 1, & t = T \end{cases} \tag{21}$$

- $\gamma_t\left(y_1^t, i\right)$: probabilidade de observar a sequência de observação parcial y_1^t ao longo do melhor caminho que passa pelo estado *i* no instante *t*.

$$\gamma_t\left(y_1^T, i\right) \equiv \begin{cases} P\left(X_0 = i\right), & t = 0 \\ \max\limits_{i_0^{t-1}} P\left(X_0^{t-1} = i_0^{t-1}, X_t = i, Y_1^t = y_1^t\right), & t > 0 \end{cases} \tag{22}$$

As variáveis α e β podem ser utilizadas para calcular a probabilidade de emissão total $P\left(y_1^T \mid W\right)$ através das expressões

$$\begin{aligned} P\left(Y_1^T = y_1^T\right) &= \sum_i \alpha_T\left(y_1^T, i\right) \\ &= \sum_i \pi_i \beta_0\left(y_1^T, i\right) \end{aligned} \tag{23}$$

Uma aproximação para calcular esta probabilidade consiste em seguir somente o caminho de máxima probabilidade. Isto pode ser feito utilizando-se a quantidade γ :

$$P\left(Y_1^T = y_1^T\right) = \max_i \gamma_T\left(y_1^T, i\right) \tag{24}$$

Esta aproximação corresponde ao algoritmo de Viterbi.

O cálculo das probabilidades acima é realizado em uma estrutura em forma de treliça, mostrada na Figura 4. Por simplicidade, pode-se assumir na figura que o HMM representa uma palavra e que o sinal de entrada corresponde à pronúncia de uma única palavra.

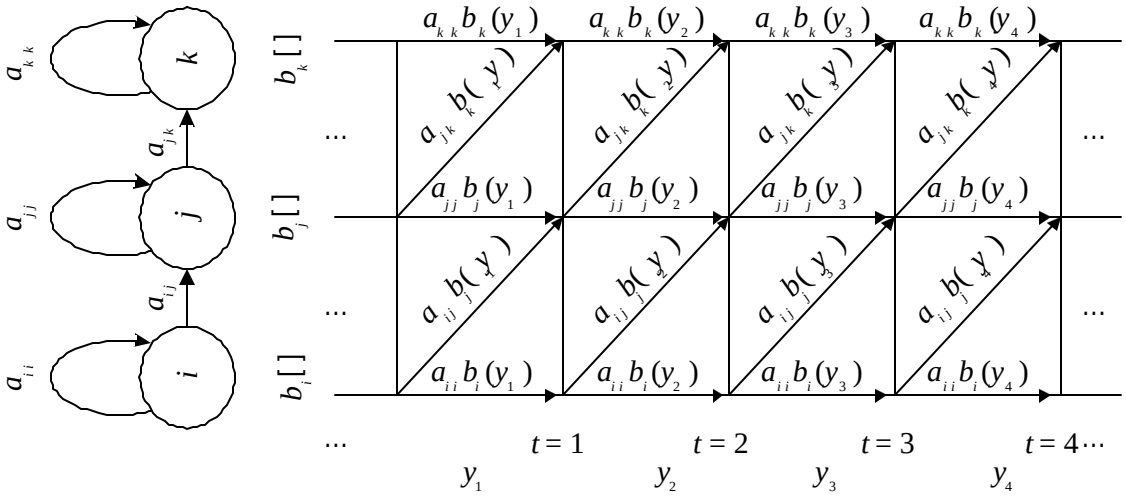


Figura 4: Exemplo de funcionamento do algoritmo de Viterbi.

Cada coluna da treliça armazena os valores das verossimilhanças acumuladas em cada estado do HMM para todos os instantes de tempo, e todo intervalo entre duas colunas consecutivas corresponde a um quadro[3] de entrada.

As setas na treliça representam transições no modelo que correspondem a possíveis caminhos no modelo do instante inicial até o final. O cálculo é realizado por colunas, atualizando as probabilidades dos nós a cada quadro, utilizando fórmulas de recursão as quais envolvem os valores de uma coluna adjacente, as probabilidades de transição dos modelos, e os valores das densidades de saída para o quadro correspondente. Para os coeficientes γ, o cálculo começa na primeira coluna à esquerda, cujos valores iniciais são dados por π_i, e termina na última coluna à direita, com a probabilidade final dada pela equação (20).

[3] Um quadro é definido como o intervalo de tempo em que é gerado um vetor de parâmetros acústicos. Valores típicos estão entre 10 e 20 ms.

O algoritmo usado para calcular os coeficientes γ é conhecido como *algoritmo de Viterbi*, e pode ser visto como uma aplicação de programação dinâmica para encontrar o caminho de máxima verossimilhança em um grafo. A fórmula de recursão é dada por:

$$\gamma_t(\mathbf{y}_1^T, i) = \begin{cases} \pi_i, & t = 0 \\ \max_j \gamma_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_i(y_t), & t > 0 \end{cases} \tag{25}$$

Monitorando o estado j que fornece a maior probabilidade na fórmula de recursão acima, é possível, no final da sequência de entrada, recuperar a sequência de estados visitada pelo melhor caminho, realizando então um tipo de alinhamento temporal dos quadros de entrada com os estados do modelo.

Todos estes algoritmos têm uma complexidade $O(MT)$, onde M é o número de transições não nulas e T o comprimento da sequência de entrada. M pode ser no máximo igual a S^2, onde S é o número de estados no modelo, mas é geralmente muito menor, uma vez que a matriz de probabilidades de transição é geralmente esparsa. De fato, nos modelos *left-right*, uma escolha comum é fazer $a_{ij} = 0,\ j - i > 2$, como no caso ilustrado na Figura 2.

Geralmente, o reconhecimento é baseado em um processo de busca que leva em conta todas as segmentações possíveis da sequência de entrada em palavras, e as probabilidades a priori que o modelo de linguagem associa a sequências de palavras. Podem ser obtidos bons resultados com modelos de linguagem simples tais como probabilidades bigrama ou trigrama [13].

4.4.1. Viterbi Beam Search.

O tamanho do espaço de busca cresce de acordo com o número de palavras no vocabulário. Para sistemas de ditado, onde são comuns vocabulários de dezenas de

milhares de palavras, o espaço de busca torna-se tão grande que o custo computacional torna-se proibitivo. Entretanto a distribuição irregular das probabilidades nos diferentes caminhos pode ajudar. O que acontece é que, quando o número de estados é grande, em cada instante de tempo, uma grande parte destes estados têm uma verossimilhança acumulada que é muito menor do que a verossimilhança máxima, de modo que é bastante improvável que um caminho que passe por um destes estados venha a ser o melhor ao final da locução.

Esta consideração leva a uma técnica de redução da complexidade chamada de *Beam Search* [15], que consiste em desprezar, em cada instante de tempo, os estados cuja verossimilhança acumulada seja menor do que a verossimilhança máxima menos um dado limiar. Desta maneira, os cálculos necessários para expandir nós ruins são evitados. Está claro pela natureza do critério de poda desta técnica de redução que ela pode causar a perda do melhor caminho. Na prática, uma boa escolha do limiar de poda resulta em um ganho de velocidade de uma ordem de magnitude, introduzindo uma quantidade desprezível de erros de busca.

5. Algoritmos de Busca.

5.1. Introdução.

O reconhecimento de fala contínua difere do reconhecimento de palavras isoladas no modo com que o usuário deve pronunciar as palavras. No reconhecimento de palavras isoladas é necessário que o locutor efetue pausas breves entre as palavras de modo que o sistema possa determinar as fronteiras entre estas de forma precisa. Já em fala contínua, o locutor pode falar de modo natural, sem efetuar pausas entre as palavras. Neste caso, a determinação das fronteiras entre as palavras e consequentemente do número de palavras na locução fica a cargo do sistema de reconhecimento.

A premissa básica do reconhecimento de fala contínua é que o reconhecimento é baseado em modelos de palavras (possivelmente formadas a partir da concatenação de sub-unidades fonéticas para os casos de grandes vocabulários). Uma vez definidos os modelos das palavras, o problema do reconhecimento resume-se em encontrar a sequência ótima (concatenação) de modelos de palavras que combine melhor (em um sentido de máxima verossimilhança) com a locução desconhecida.

Uma grande variedade de aproximações, todas baseadas na técnica de programação dinâmica, foram propostas e avaliadas [50][5][6][32][4][43][35][36][33].

O primeiro algoritmo para o reconhecimento de palavras conectadas foi proposto por Vintsyuk [50] que mostrou como as técnicas de programação dinâmica poderiam ser utilizadas para descobrir a sequência de palavras ótima que combina com uma dada

locução de entrada. O procedimento de Vintsyuk processa o sinal de fala de maneira síncrona, e portanto o seu trabalho pioneiro formou a base para várias soluções baseadas em programação dinâmica para os problemas de reconhecimento de fala.

Várias outras estruturas de busca baseadas em programação dinâmica foram propostas para resolver o problema de reconhecimento de fala, baseadas na concatenação de modelos de palavras e sub-unidades, incluindo a aproximação de sistemas estatísticos de Baker desenvolvido na Carnegie Mellon University [5][6] (a qual foi seguida pela pesquisa de Lowerre na mesma instituição [32]), a aproximação estatística dos pesquisadores da IBM [4], e vários algoritmos de casamento de modelos de palavras [43][35][36][33].

A maior contribuição destas pesquisas iniciais é a idéia de representar todas as fontes de conhecimento usadas no reconhecimento (representação das palavras, modelo de linguagem, etc.) como redes (tanto determinísticas como estocásticas) que poderiam ser facilmente integradas com a rede básica que representa as unidades básicas (palavras ou sub-unidades fonéticas). A busca poderia então ser realizada eficiente e acuradamente utilizando técnicas de programação dinâmica. Foram propostos vários algoritmos para encontrar o melhor caminho através de uma rede: o S*tack Algorithm* desenvolvido por Jelinek [24], o *Two Level* de Sakoe [43], o *Level Building* de Myers e Rabiner [35], o *One Step* de Ney [36], entre outros. Estes algoritmos são capazes de obter a melhor sequência de palavras que combina com uma dada locução de entrada, sujeita a uma grande variedade de limitações sintáticas (modelos de linguagem).

5.2. Reconhecimento de fala contínua via decodificação de rede finita de estados.

Como dito anteriormente, o problema do reconhecimento de fala pode ser organizado em uma hierarquia de redes de estados finitos com um número finito de nós e arcos correspondentes às fontes de conhecimento acústico, fonético e sintático e suas

interações. Em um sistema de reconhecimento de fala, o conhecimento acústico está relacionado à forma de parametrização do sinal de voz (parâmetros LPC, cepstrais, etc.), o conhecimento fonético, à transcrição fonética das palavras do vocabulário, e o conhecimento sintático, ao modelo de linguagem. O reconhecimento de uma locução corresponde a encontrar o caminho ótimo através da rede de estados finitos.

Esta busca pode ser realizada através de decodificação sequencial usando os conceitos de programação dinâmica e o princípio da otimalidade definido por Bellman [8]: “um conjunto de decisões ótimas tem a propriedade de, qualquer que tenha sido a primeira decisão, as decisões restantes precisam ser ótimas em relação à saída da primeira decisão”. Em termos do problema de encontrar o melhor caminho através de uma rede de estados finita, o princípio da otimalidade permite que a decodificação seja feita de modo síncrono, pois toda a informação requerida para os caminhos ótimos locais está disponível, e os caminhos ótimos globais podem ser encontrados a partir dos caminhos ótimos locais.

Para a tarefa de reconhecimento de palavras conectadas, é conveniente decompor a rede em dois níveis: nível de frases (gramático) e nível intra-palavra. Cada um dos níveis tem propriedades completamente diferentes. O nível intra-palavra é geralmente um modelo de palavra, que pode ser um HMM da palavra inteira, ou uma representação da palavra formada pela concatenação de modelos HMM de sub-unidades acústicas.

O nível gramático é representado por uma rede gramática (de acordo com o modelo de linguagem), na qual os nós representam fronteiras de palavras, e os arcos representam modelos de palavras. Estas representações vão desde redes simples com poucas restrições sintáticas (por exemplo, gramáticas bigrama ou trigrama) a redes gramáticas altamente complexas e restritivas (por exemplo, gramáticas sensíveis a contexto).

Para realizar a busca em uma rede de estados finita é necessário estabelecer uma medida de custo (por exemplo, distância, verossimilhança) associada ao caminho. Esta medida inclui o custo de estar em um nó intra-palavra, o custo de fazer transições de um nó intra-palavra a outro, e o custo de entrar em um arco gramático. Na tarefa de reconhecimento de fala, na qual os modelos das palavras são caracterizados por um

HMM (ou concatenação de HMM's), o custo acumulado de um caminho que passa por um determinado nó na rede de estados finita no instante t pode ser definida como o negativo da verossimilhança acumulada do caminho no instante t. Esta verossimilhança é definida como o logaritmo da probabilidade daquele caminho. Assim, a rede resultante é uma rede de estados finita estocástica onde o custo de um caminho depende da sequência de observação, do tempo que o sistema ficou em determinado nó, e da história de transições do caminho.

O custo de estar em um nó interno no instante t é relacionado à probabilidade de observar o vetor acústico naquele estado, no instante t, e pode ser definido como o negativo do logaritmo da probabilidade da observação no estado. O custo de fazer uma transição interna inclui o negativo do logaritmo da probabilidade de transição, mais alguma possível penalidade de duração de estados, que depende do tempo em que o sistema permaneceu naquele estado. Finalmente, o custo de deixar o nó gramático esquerdo de um arco gramático inclui uma possível penalização de transição de palavra. Com todos os custos atribuídos convenientemente, o procedimento de busca pelo melhor caminho na rede de estados finita é essencialmente o mesmo de encontrar o caminho de custo mínimo através da rede, ou equivalentemente, realizar uma decodificação de máxima verossimilhança.

5.3. Definição do problema.

Seja $x(t)$ um sinal de voz digitalizado. A intervalos regulares de tempo, tipicamente a cada 10 ou 20 ms, é calculado um vetor de parâmetros acústicos y_t. As sequências de vetores de parâmetros acústicos são tratadas como observações dos modelos das palavras, usados para calcular $P\left(y_1^T \mid W\right)$, a probabilidade de observar a sequência y_1^T de vetores quando se pronuncia uma sequência de palavras *W*.

Dada uma sequência y_1^T, o sistema de reconhecimento de fala gera uma sequência $\hat{W}$ de palavras, através de um processo de busca dado pela regra:

$$\hat{W} = \arg\max_{W} P\left(y_1^T \mid W\right)P(W) \tag{26}$$

onde $\hat{W}$ corresponde à sequência de palavras que apresentou a máxima probabilidade a posteriori (MAP). $P\left(y_1^T \mid W\right)$ é calculado a partir de *modelos acústicos*, enquanto que $P(W)$ é calculado a partir de *modelos de linguagem*.

Neste trabalho, as palavras são caracterizadas por modelos HMM's os quais, por sua vez, são formados pela concatenação dos modelos HMM's das sub-unidades fonéticas de sua transcrição fonética.

Das várias técnicas propostas para a decodificação, duas foram estudadas e implementadas neste trabalho: o *Level Building* de Myers e Rabiner [35] e o *One Step* de Ney [36]. O algoritmo *One Step* diferencia-se do *Level Building* na forma de implementação: enquanto o *Level Building* é síncrono por palavras, o *One Step* é síncrono por quadros.

O *Level Building* teve uma grande importância histórica na redução da complexidade dos cálculos necessários ao reconhecimento de fala contínua. Entretanto, com o advento de máquinas mais poderosas, que permitiram o reconhecimento de fala contínua em tempo real, esta abordagem passou a ser inviável pois, como é síncrona por palavra, tem que esperar até o final da locução para iniciar os processamentos, o que não acontece com o *One Step* por ser síncrono com o tempo. Entretanto, em termos de resultados da decodificação, ambos são equivalentes.

5.3.1. Level Building.

Seja λ o conjunto de V modelos HMM das palavras do vocabulário de reconhecimento, e λ^v, $1 \le v \le V$, cada um dos modelos de palavras deste vocabulário, possivelmente formadas a partir da concatenação de modelos HMM de sub-unidades fonéticas. Para achar a sequência ótima de HMM's que melhor combine com a

sequência de observação *O* (maximize a verossimilhança), utiliza-se um processo de busca baseado no algoritmo de Viterbi.

Para cada modelo HMM de palavra λ^v e, a cada nível *l*, faz-se uma busca de Viterbi contra *O*, iniciando no quadro 1 no nível 1, e armazena-se para cada quadro *t* os seguintes valores:

1. $P_l^v(t)$, $1 \le t \le T$, verossimilhança acumulada do quadro *t*, no nível *l*, para a palavra de referência λ^v, ao longo do melhor caminho.
2. $B_l^v(t)$, $1 \le t \le T$, ponteiro que indica onde o caminho se iniciou no começo do nível.

Ao final de cada nível *l* (onde o nível corresponde à posição da palavra na sequência de palavras), realiza-se uma maximização sobre *v* para obter o melhor modelo em cada quadro *t* da seguinte maneira:

$$P_l^B(t) = \max_{1 \le v \le V} P_l^v(t), 1 \le t \le T \tag{27}$$

$$W_l^B(t) = \arg\max_{1 \le v \le V} P_l^v(t), 1 \le t \le T \tag{28}$$

$$B_l^B(t) = B_l^{W_l^B(t)}(t), 1 \le t \le T \tag{29}$$

onde:

$W_l^B(t)$: é o modelo da palavra que obteve a maior verossimilhança no quadro *t*, nível *l*.

$B_l^B(t)$: armazena o ponteiro do modelo da palavra vencedora.

Cada novo nível começa com a maior verossimilhança do quadro anterior do nível anterior e incrementa o valor da verossimilhança combinando os modelos das palavras que começam no quadro inicial. Este processo é repetido através de um número de níveis equivalente ao número máximo de palavras esperado para uma dada locução.

Ao final de cada nível, a melhor sequência de palavras de comprimento l ($1 \leq l \leq L$) com probabilidade $P_l^B(T)$ é obtida usando o vetor $B_l^B(t)$. A melhor sequência de palavras é a de máximo $P_l^B(T)$ sobre todos os níveis l. O algoritmo *Level Building* é ilustrado na figura abaixo:

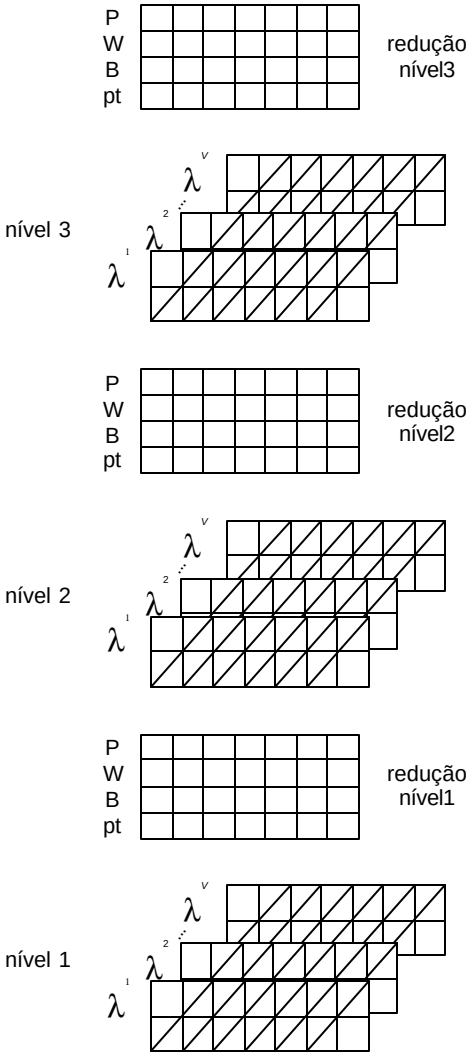


Figura 5: Exemplo de funcionamento do algoritmo *Level Building*.

5.3.2. One Step.

O algoritmo *One Step* realiza os mesmos cálculos do *Level Building*, com a diferença de que o processamento é feito por quadros e não por palavras. Esta diferença sutil é bastante poderosa, pois oferece a oportunidade de processamento em tempo real, coisa que não é possível com o algoritmo *Level Building*. Pode-se ver o *One Step* como um algoritmo 'transposto' em relação ao *Level Building*. No desenvolvimento a seguir, isto ficará mais claro.

Para cada nó na rede de estados finita, em todo instante t, o algoritmo procura pelo melhor caminho que chega ao nó naquele instante, e constrói o caminho ótimo de duração t para aquele nó a partir de todos os caminhos de duração $(t-1)$. O princípio da otimalidade da programação dinâmica garante que o melhor caminho para qualquer nó i, no instante t, pode ser determinado a partir dos melhores caminhos para todos os nós j, no instante $(t-1)$, mais o custo de fazer a transição do nó j para o nó i no instante t.

Neste trabalho foi utilizada uma versão baseada no trabalho de Lee & Rabiner [28], e o algoritmo é apresentado a seguir:

1. Inicialização de todos os nós e variáveis de armazenamento dos caminhos
2. Construção do caminho ótimo quadro a quadro:

Para todo quadro de entrada faça
  Para todo nó gramático faça
    Para toda palavra do vocabulário faça
      Calcule verossimilhanças dos nós intra-palavra (algoritmo de Viterbi)
      Aplique penalização de duração
      Atualize verossimilhanças e informações sobre o caminho ótimo
    Próxima palavra
    Determine palavra vencedora no nó gramático (redução de nível)
    Atualize verossimilhanças e informações do caminho para o nó gramático
  Próximo nó gramático
Próximo quadro de entrada

3. Recuperação da sequência de palavras
Para cada nó terminal ativo na rede faça
Recupere o caminho ótimo para identificar a sequência de palavras reconhecida
Próximo nó terminal ativo na rede
4. Determine a sequência de palavras reconhecida.

Pode-se ver que existem 3 laços principais: o mais externo em relação aos quadros de entrada, outro intermediário, relacionado aos níveis e, finalmente o mais interno, relacionado às palavras do vocabulário.

Neste algoritmo, as seguintes quantidades são armazenadas a cada quadro t:

Estruturas intra-palavra:

- $like(i, j, k)$: verossimilhança do estado k da palavra j no nível gramático i.
- $elapse(i, j, k)$: duração do melhor caminho desde o início até o instante atual para o estado k da palavra j no nível gramático i.

Estruturas gramaticais:

- $glike(i, t)$: verossimilhança do melhor caminho que chega ao nó gramático i no instante t.
- $word(i, t)$: palavra vencedora para o nó gramático i, no instante t.
- $bp(i, t)$: instante em que o caminho que chegou ao nó gramático i, no instante t se iniciou
- $prob\_ant(i, t)$: probabilidade de transição a partir do nó gramático i, no instante t.

Para atualização das estruturas intra-palavra é utilizado o algoritmo de Viterbi. Neste é utilizado um vetor temporário, aqui denominado $scratch(i)$. O algoritmo é descrito a seguir, e ilustrado na Figura 6:

1.Inicialização:

$glike(0,t)=0$

$like(i,\ j,0)=glike(i-1,t-1)$

$elapse(i,\ j,0)$

2. Calcular o melhor caminho que chega ao nó $(i,\ j,k)$, $1\le k\le N$, no instante t, onde N é o número de estados da palavra em consideração:

$scratch(k)=like(i,\ j,k)+a_{k,k}$

$scratch(k-1)=like(i,\ j,k-1)+a_{k-1,k}$

$se\ (scratch(k)>scratch(k-1))$

$elapse(i,\ j,k)=elapse(i,\ j,k)+1$

senão

$scratch(k)=scratch(k-1)$

$elapse(i,\ j,k)=elapse(i,\ j,k-1)+1$

fim

3. Atualizar as verossimilhanças acumuladas nos estados k, $1\le k\le N$:

$like(i,\ j,k)=scratch(k)+b_k\ (t)$

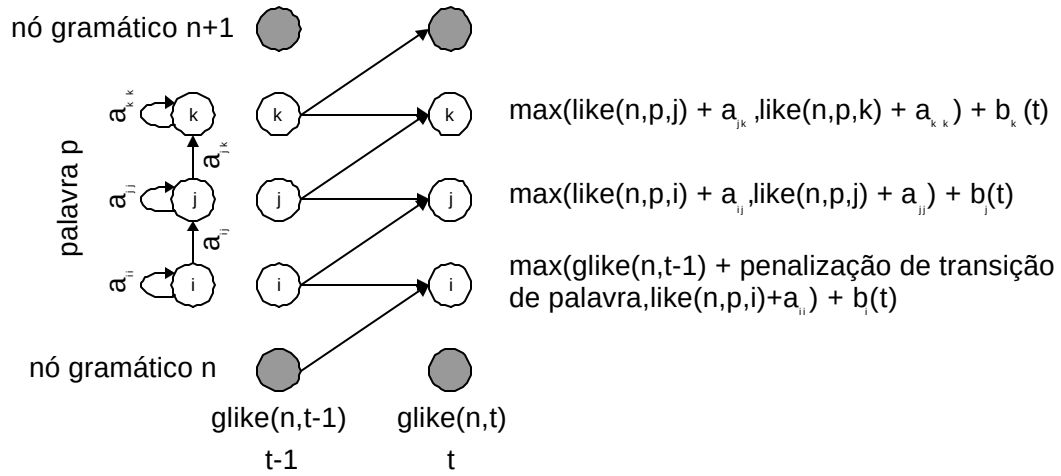


Figura 6: Ilustração do funcionamento do algoritmo de Viterbi na implementação do algoritmo *One Step*.

A atualização das estruturas gramaticais, em cada nível *i*, e em cada instante de tempo *t*, é realizada pelo algoritmo de fusão de caminhos. Algumas variáveis adicionais são utilizadas:

máximo: maior verossimilhança dentre todos os caminhos que chegam ao nó gramático

n_palavras: número de palavras consideradas no nível atual

n_estados(*j*): número de estados do modelo da palavra *j*.

O algoritmo para atualização das estruturas gramaticais fica:

$máximo = -\infty$

$para\ \ j = 1\ até\ n\_palavras$

$\quad se\ (like(i, j, k) > máximo)$

$\quad\quad máximo = like(i, j, k)$

$\quad\quad word(i, t) = j$

$\quad fim$

próximo j

$glike(i, t) = máximo$

Finalmente, a sequência de palavras reconhecida é recuperada através do algoritmo de *backtracking*. As variáveis adicionais neste caso são:

T: último quadro da locução a ser reconhecida.

máximo: maior verossimilhança entre todos os níveis, no instante final *T*.

quantas: número de palavras reconhecidas no processo de decodificação

$palavra(n)$: armazena as palavras da sequência reconhecida

$duração(n)$: armazena a duração de cada palavra da sequência reconhecida, em quadros

% Determinando número de palavras na locução:

$máximo = -\infty$

Para todo nível faça

$se\,(glike(nível,T) > máximo)$

$máximo = glike(nível,t)$

$quantas = nível$

fim

Próximo nível

% Determinando sequência de palavras reconhecida

$n = quantas$

$t = T$

$enquanto\,(t \geq 0)$

$palavra(n) = word(n,t)$

$duração(n) = bp(n,t)$

$t = t - duração(n)$

$n = n - 1$

fim

Ao final deste procedimento, as palavras reconhecidas são armazenadas no vetor *palavra*, em ordem invertida. As durações de cada uma delas é armazenada no vetor *duração.*

É importante frisar que, em termos de resultados, o *Level Building* e o *One Step* são algoritmos equivalentes. Entretanto, na implementação, o *One Step* proporciona facilidades como a possibilidade de processamento em tempo real e a redução de complexidade através do procedimento *Beam Search*.

5.4. Inclusão do modelo de duração de palavras.

Na determinação do caminho ótimo através do algoritmo de Viterbi, tanto no caso do *Level Building* como no *One Step*, pode-se associar uma duração a cada palavra,

em cada nível de busca. Para os HMM's, a probabilidade de duração $P_i(d)$ associada ao estado S_i com probabilidade de auto-transição a_{ii} é dada por:

$$P_i(d) = (a_{ii})^{d-1}(1 - a_{ii}) \tag{30}$$

A quantidade $P_i(d)$ pode ser vista como a probabilidade de d observações consecutivas no estado S_i. Este modelo de duração exponencial é bastante inadequado para representar sinais reais, podendo fazer com que a duração encontrada esteja muito distante de uma duração 'média' atribuída à locução daquela palavra. Pode-se modelar explicitamente a densidade de duração através de formas explicitamente analíticas, mas o custo computacional é elevadíssimo [40].

Uma forma alternativa proposta por Rabiner [40] associa à duração d de cada palavra i do vocabulário uma função densidade de probabilidade gaussiana $f_i(d)$

$$f_i(d) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(- \frac{(d - \bar{d}_i)^2}{2\sigma_i^2} \right) \tag{31}$$

onde $\bar{d}_i$ e σ_i^2 são, respectivamente, a média e a variância da duração da palavra i. Estes valores são obtidos a partir da segmentação das locuções de treinamento.

O procedimento para incorporar o modelo de duração aos algoritmos de busca é o seguinte:

A cada instante t, determina-se a duração da palavra i, no nível l, através da recuperação do caminho ótimo determinado pelo algoritmo de Viterbi.

$d_i(t) = t - B_l^i(t)$, para o algoritmo level building (32)

$$d_i(t) = t - elapse(l, i, n\_estados(j)), \text{ para o algoritmo one step} \quad (33)$$

A verossimilhança acumulada para uma dada palavra é penalizada de acordo com a função densidade de probabilidade gaussiana, com os parâmetros da palavra em análise, no ponto determinado por $d_i(t)$:

$$P_l^i(t) = P_l^i(t) + \log_{10}(f_i(d_i(t))), \text{ level building} \quad (34)$$

$$like(l, i, n\_estados(j)) = like(l, i, n\_estados(j)) + \log_{10}(f_i(d_i(t))), \text{ one step} \quad (35)$$

Embora claramente heurístico, este método proporcionou uma melhora significativa em testes anteriores realizados com uma base de dados dependente de locutor [33].

No presente trabalho, o modelo de duração de palavras foi levantado manualmente a partir das locuções de um único locutor (o mesmo utilizado nos testes com dependência de locutor, cujos resultados são mostrados no Capítulo 7). Nos testes com independência de locutor, sempre haverá casos em que as durações das palavras nas locuções de teste estejam significativamente distantes daquelas armazenadas no modelo de duração. Isto pode fazer com que o reconhecimento seja prejudicado nestes casos, mas algum procedimento de adaptação poderia minimizar este problema.

5.5. Inclusão do modelo de linguagem.

No sistema foi utilizado um modelo de linguagem do tipo pares-de-palavras, que é uma simplificação do modelo *bigrama*, descrito no Capítulo 2. Esta aproximação pode ser descrita através da expressão:

$$\tilde{P}(W) \approx \prod_{i=1}^{n} G(w_i \mid w_{i-1}) \tag{36}$$

onde

$$G(w_i \mid w_{i-1}) = \begin{cases} 1, P(w_i \mid w_{i-1}) \neq 0 \\ 0, P(w_i \mid w_{i-1}) = 0 \end{cases} \tag{37}$$

Este modelo de linguagem pode ser visto como uma versão determinística do modelo bigrama. A escolha por este modelo em detrimento do bigrama é devida à limitação da base de dados: como é muito pequena, a utilização das frequências bigrama poderia polarizar o algoritmo de busca em alguns casos. Por exemplo, supondo que a sequência “a casa” tenha ocorrido duas vezes, e a sequência “a taça” apenas uma vez, o sistema poderia reconhecer a locução “a taça” como “a casa”, visto que são parecidas, e o modelo de linguagem atribuiria uma probabilidade duas vezes maior para a sequência “a casa”.

A incorporação do modelo de linguagem aos algoritmos de busca é trivial: ao início de cada nível, em cada instante t, verifica-se qual a palavra vencedora no nível anterior e, se a palavra sob análise for permitida pelo modelo de linguagem, é expandida.

6. Sistema Desenvolvido.

O sistema desenvolvido é formado por três módulos principais:

1) Módulo de extração de parâmetros e quantização vetorial.
2) Módulo de treinamento
3) Módulo de geração de modelo de linguagem
4) Módulo de reconhecimento

O primeiro módulo é formado por dois programas: o programa de extração de parâmetros, que converte um sinal de voz digitalizado em vetores acústicos, e o programa de quantização vetorial. O segundo é formado por quatro programas: programa de treinamento dos HMM's, programa de detecção de trifones, programa de combinação de modelos baseado no procedimento deleted interpolation, e o programa de geração de gramática bigrama. O terceiro é o responsável pela geração do modelo de linguagem, baseado no modelo de gramática bigrama. Por fim, o último módulo é formado pelo programa de reconhecimento.

Os programas foram implementados em linguagem C++ para a plataforma Windows. Na implementação, teve-se o cuidado de criar uma interface visual bastante amigável e intuitiva, um código estruturado e extensamente documentado, de forma que outros pesquisadores possam desenvolver as suas idéias a partir deste sistema. Com isto, o tempo necessário para testar novas idéias ficou bastante reduzido. Neste laboratório,

este sistema já está sendo utilizado por outros pesquisadores, nas áreas de adaptação ao locutor e reconhecimento de dígitos conectados.

Nas seções seguintes o sistema será mostrado com mais detalhes.

6.1. Módulo de extração de parâmetros e quantização vetorial.

Este módulo é o responsável por transformar as locuções de entrada em parâmetros que possam ser interpretados pelos módulos seguintes. Na Figura 7 tem-se um diagrama de blocos onde é mostrada a arquitetura deste módulo.

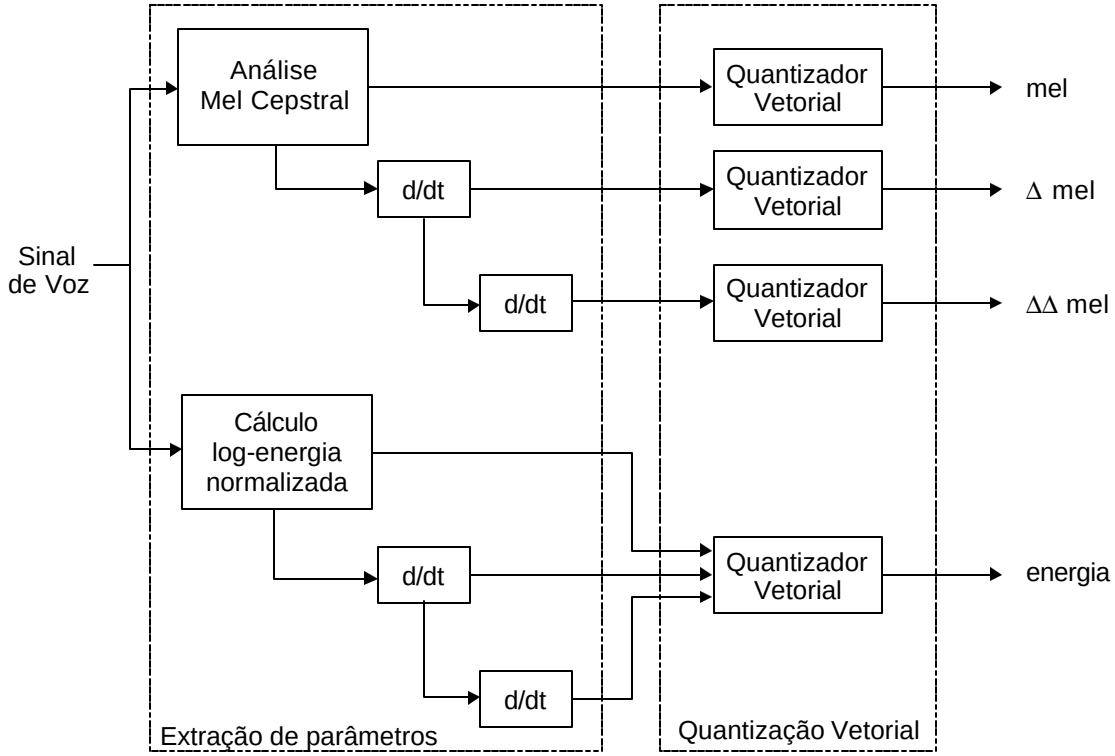


Figura 7: Diagrama de blocos do módulo de extração de parâmetros e quantização vetorial.

6.1.1. Extração de parâmetros.

Este módulo tem por entrada um sinal de voz em formato WAV ou binário, e calcula parâmetros da locução. Atualmente estão disponíveis os parâmetros mel-cepstrais de ordem 12 [14] e log-energia normalizada, bem como seus respectivos parâmetros delta e delta-delta, nas frequências de amostragem de 8, 11,025 e 16 kHz, tanto para 8 como 16 bits de resolução. Embora algumas destas opções não sejam utilizadas neste trabalho, optou-se por criar um programa mais versátil, que pudesse ser utilizado com outras bases de dados.

Os parâmetros são calculados utilizando-se janelas de 20 ms, atualizadas a cada 10 ms. Antes da extração, o sinal é submetido a alguns pré-processamentos: retirada do nível DC, pré-ênfase com um filtro passa altas ($1\text{-}0{,}95\ z^{-1}$), e janelamento através de uma janela de Hamming.

Os parâmetros log-energia foram normalizados tomando como referência o quadro de maior energia em toda a locução sob análise.

Para os parâmetros mel-cepstrais a ordem utilizada foi 12. A estes parâmetros foi aplicado o procedimento de remoção da média espectral, que consiste em calcular o vetor média $\bar{x}$ de todos os vetores acústicos que representam uma dada locução.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{38}$$

onde: N é o número de vetores acústicos;

x_i é o i-ésimo vetor acústico da locução.

Este vetor média é então subtraído de cada um dos vetores acústicos da locução, gerando uma versão modificada $\tilde{x}$ do vetor acústico x:

$$\tilde{x} = x - \bar{x} \tag{39}$$

Abaixo, tem-se um diagrama de blocos para este procedimento:

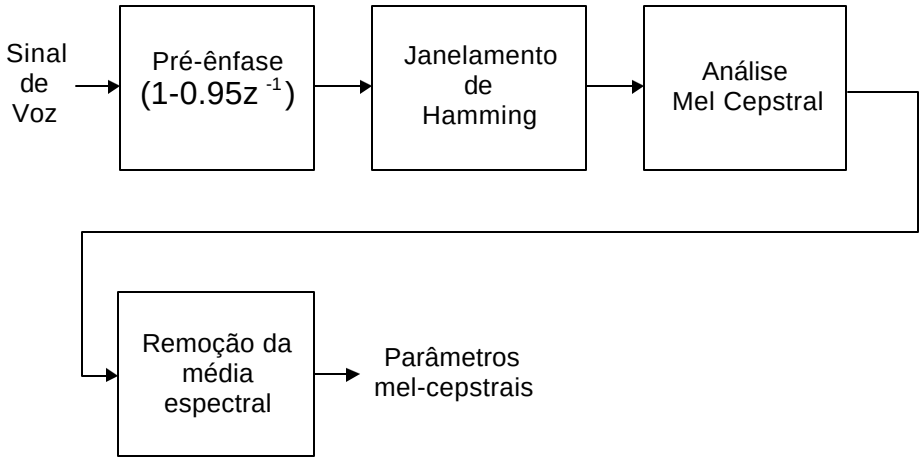


Figura 8: Diagrama de blocos do processo de extração dos parâmetros mel-cepstrais com remoção da média espectral.

A justificativa para o procedimento pode ser resumida da seguinte maneira: na extração dos parâmetros de uma locução, o sinal de voz é segmentado em trechos curtos, geralmente entre 10 e 20 ms. Desta forma, o sistema não consegue distinguir o sinal quasi-estacionário de curto termo (sinal de voz) do sinal quasi-estacionário de longo termo (ruído ambiente e/ou característica do canal). O cálculo da média dos vetores ao longo de toda a locução traria então informações sobre o sinal quasi-estacionário de longo termo, e a sua remoção teria um efeito de minimizar a influência deste sinal de longo termo no sinal de voz. Testes preliminares mostraram uma melhoria no desempenho do sistema utilizando este procedimento [53].

O sinal parametrizado é armazenado em arquivo de mesmo nome do original, mas com extensão diferente (‘.mel’ para parâmetros mel-cepstrais e ‘.ene’ para parâmetros log-energia normalizada).

Os parâmetros delta foram calculados segundo a expressão:

$$\Delta_i(n) = \frac{1}{2K+1} \sum_{k=-K}^{K} k y_{i-k}(n) \qquad (40)$$

onde:

$y_i(n)$: é o *n*-ésimo elemento do vetor de parâmetros y_i;

$\Delta_i(n)$: é o *n*-ésimo elemento do vetor delta correspondente ao vetor de parâmetros y_i;

K : é o número de quadros adjacentes de vetores de parâmetros a serem considerados no cálculo dos parâmetros delta. Neste trabalho foi utilizado $K = 1$ tanto para o cálculo dos parâmetros delta como para os delta-delta.

6.1.2. Quantizador Vetorial.

Como o sistema de reconhecimento é baseado em modelos de Markov discretos, torna-se necessário quantizar os parâmetros de entrada através de um quantizador vetorial.

O sistema de quantização é formado por dois módulos: um módulo de treinamento, responsável pela geração dos vetores código do quantizador e outro responsável pela quantização propriamente dita.

Para a parte de treinamento foi utilizado o algoritmo LBG em sua versão *splitting* [31] para gerar os vetores código do quantizador. No presente trabalho foram utilizados dicionários de códigos de 256 vetores para cada um dos parâmetros de entrada. Estes dicionários foram gerados a partir das locuções de treinamento.

A quantização foi realizada através de comparação exaustiva de cada vetor de entrada com cada um dos 256 vetores do dicionário de códigos, utilizando como medida de distorção a distância euclidiana:

$$d(x, x_i) = \sqrt{(x - x_i)(x - x_i)^t} \tag{41}$$

onde:

$d(x, x_i)$: é a distância euclidiana entre os vetores x e x_i.

x: é o vetor coluna que se deseja quantizar.

x_i : é o i-ésimo vetor do dicionário de códigos.

t: indica matriz transposta.

Os parâmetros log-energia normalizada, bem como suas derivadas primeira e segunda são grandezas escalares. Ao invés de realizar uma quantização escalar para cada um destes parâmetros, optou-se por agrupá-los em um único vetor de três posições e realizar uma quantização vetorial sobre estes vetores.

6.2. Módulo de treinamento.

Este módulo é o responsável pelo treinamento dos modelos HMM das sub-unidades fonéticas a serem utilizados na etapa de reconhecimento.

O processo de treinamento das sub-unidades fonéticas é dividido em duas partes: primeiro são gerados modelos de fones independentes de contexto (os mesmos utilizados para a transcrição fonética das locuções de treinamento). Estes irão servir como modelos iniciais para o treinamento dos modelos de fones dependentes de contexto (trifones).

Foram desenvolvidos três programas para esta parte do sistema: um para o treinamento das sub-unidades acústicas, outro para detecção e inicialização dos fones dependentes de contexto, e o último responsável por mesclar os modelos de fones independentes de contexto com modelos de trifones utilizando o algoritmo *Deleted Interpolation*.

6.2.1. Programa de treinamento das sub-unidades.

Este programa tem por função treinar os modelos HMM das sub-unidades acústicas a partir de locuções de treinamento parametrizadas e quantizadas e das respectivas transcrições fonéticas. O algoritmo de treinamento utilizado é o Baum-Welch [40]. Uma visão geral da arquitetura deste programa é fornecida na Figura 9:

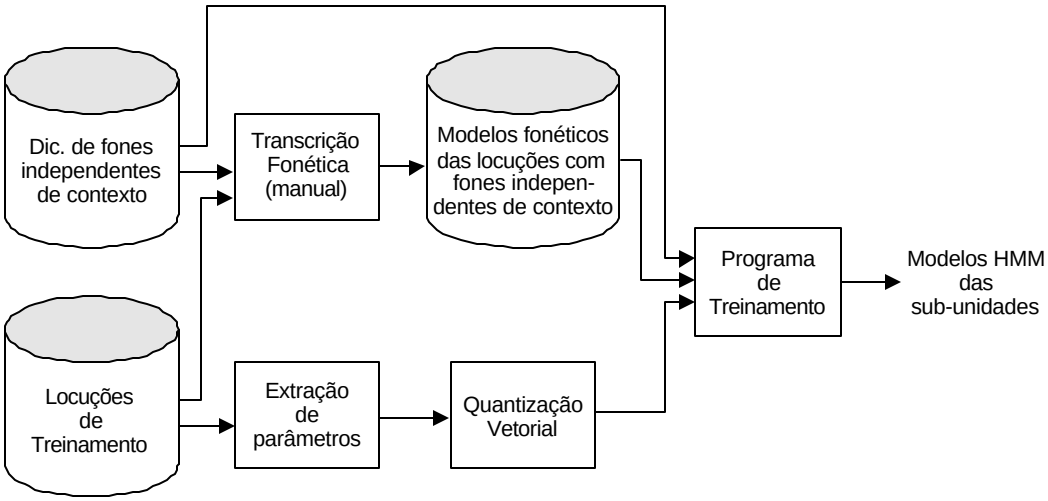


Figura 9: Esquema de funcionamento do programa de treinamento das sub-unidades com indicação das informações a serem fornecidas ao sistema.

Como mostrado na Figura 9, é necessário fornecer ao programa de treinamento as seguintes informações:

1) Sub-unidades acústicas a serem utilizadas na transcrição fonética das locuções de treinamento (fones independentes de contexto).
2) Transcrição das locuções utilizando estas sub-unidades fonéticas.
3) Locuções de treinamento parametrizadas e quantizadas.

O procedimento adotado para o treinamento das sub-unidades é o seguinte: inicialmente são criados modelos HMM para cada uma das sub-unidades acústicas. A arquitetura utilizada para os HMM's é mostrada na Figura 10.

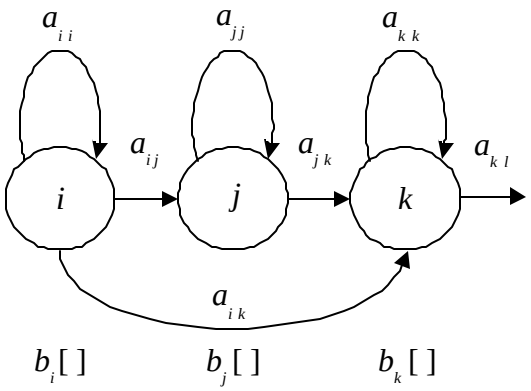


Figura 10: Modelo HMM utilizado para cada uma das sub-unidades fonéticas. A probabilidade de transição a_{kl} indica a probabilidade de fazer uma transição para a sub-unidade seguinte.

Para a inicialização destes modelos foram testadas duas abordagens: uma utilizando uma distribuição uniforme e outra utilizando o algoritmo *Segmental K-Means* [27].

Pelo método da distribuição uniforme, assume-se que inicialmente todos os símbolos são equiprováveis, e as probabilidades de emissão de símbolos de saida $b_j(y)$ são inicializadas com valor 1/*num_vet*, onde *num_vet* é o número de vetores com o qual foi feita a quantização vetorial dos parâmetros acústicos (256 neste trabalho). As probabilidades de transição são inicializadas como sendo equiprováveis, como mostrado na Figura 11.

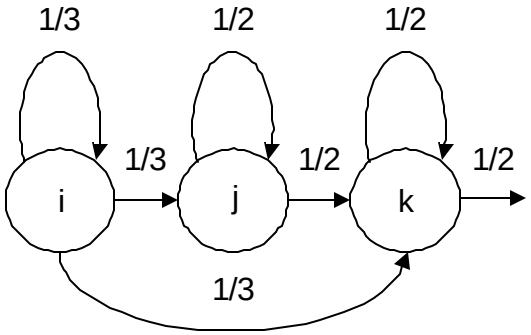


Figura 11: Valores iniciais para as probabilidades de transição dos modelos dos fones para inicialização com distribuição uniforme.

O método via *Segmental K-Means* é dividido em duas partes: inicialização e pré-treinamento. Na etapa de inicialização, as locuções de treinamento são divididas em *m* partes iguais (de mesmo comprimento), sendo *m* o número de sub-unidades fonéticas da transcrição fonética multiplicada pelo número de estados de cada modelo HMM (3 neste trabalho). É criado um modelo HMM para a locução concatenando-se os modelos HMM das sub-unidades acústicas referentes à sua transcrição fonética. Faz-se então uma contagem dos símbolos que ocorreram em cada uma destas partes, e estas contagens, depois de transformadas em medidas de probabilidade, serão os valores iniciais de cada estado dos modelos HMM correspondentes. As probabilidades de transição são inicializadas como no caso anterior (ver Figura 11). Um exemplo ajuda a compreender melhor este procedimento:

Seja uma locução consistindo apenas da palavra ‘banana’. Incluindo os silêncios inicial e final, a transcrição para esta locução seria:

b a n an n a

Temos então 8 sub-unidades acústicas a serem treinadas, e como cada uma é modelada por um HMM de 3 estados, temos um total de 24 fdp’s a serem estimadas. Supondo que esta locução foi parametrizada com 240 quadros, teríamos 10 quadros para cada estado. Os 10 primeiros símbolos irão inicializar o primeiro estado da primeira sub-unidade acústica (primeiro estado do silêncio (#) no exemplo), os 10 seguintes o segundo, e assim por diante. A inicialização é feita por simples contagem: verifica-se quantas vezes cada um dos símbolos ocorreu nestes 10 quadros, atualizando as contagens destes símbolos nas fdp’s discretas correspondentes.

É interessante verificar que no exemplo dado, existem dois exemplares dos fones [#], [a] e [n]. Neste caso, as contagens de cada um deles é acumulada na mesma fdp. Similarmente, se tivermos mais locuções de treinamento, as contagens dos fones vão sendo acumuladas na mesma fdp. Ao final, estas contagens são transformadas em medidas de probabilidade.

Mesmo com vários exemplos de treinamento para cada fone, é muito comum a ocorrência de valores nulos para algumas posições destas fdp's discretas. O efeito de valores nulos no processo de reconhecimento é devastador, e um dos métodos empregados para evitar este inconveniente é substituir estes valores nulos por um valor pequeno. Isto equivale a dizer que a ocorrência do símbolo ao invés de ser impossível, é improvável, uma condição bem menos drástica.

Depois da inicialização dos modelos faz-se o pré-treinamento utilizando o algoritmo de Viterbi, que corresponde ao procedimento *Segmental K-Means.* O procedimento é parecido com o da inicialização descrito acima, com a diferença de que agora a segmentação não é uniforme, ou seja, a cada um dos estados são associados mais ou menos quadros dependendo do caminho escolhido pelo algoritmo de Viterbi. As probabilidades de emissão são atualizadas, como no caso anterior, pela contagem dos símbolos emitidos em cada estado, e as de transição, pelo número de quadros que o sistema ficou em cada estado.

Os testes realizados com ambas as inicializações não mostraram diferenças significativas entre um e outro procedimento, de modo que o primeiro procedimento foi adotado, por ser mais simples.

Após a inicialização vem o treinamento propriamente dito, onde é utilizado o algoritmo Baum-Welch. O procedimento é similar ao da inicialização utilizando o método *Segmental K-Means*: para cada locução de treinamento é gerado um modelo HMM através da concatenação dos modelos referentes às sub-unidades acústicas da sua transcrição fonética. Este modelo composto pode então ser tratado como uma única palavra, e a locução da frase, a palavra correspondente a este modelo composto. Desta forma o algoritmo de treinamento maximiza a probabilidade de o modelo composto gerar a locução correspondente. Depois disso, os modelos individuais das sub-unidades fonéticas são separados, e as contagens geradas pelo algoritmo Baum-Welch são acumuladas durante todo o processo de treinamento, e somente após serem processadas todas as locuções de treinamento (uma época de treinamento), são transformadas em medidas de probabilidade.

Após cada época de treinamento, faz-se uma verificação da convergência da seguinte maneira: para cada locução de treinamento monta-se o modelo HMM correspondente através da concatenação dos modelos das sub-unidades fonéticas e aplica-se o algoritmo de Viterbi para calcular a probabilidade de o modelo gerar a locução correspondente. Repetindo este procedimento para todas as locuções de treinamento, pode-se calcular uma 'probabilidade média' de os modelos gerarem as sequências de vetores acústicos correspondentes às locuções de treinamento. A cada época esta probabilidade média cresce até que um patamar é atingido. O treinamento é realizado até que a probabilidade média pare de crescer.

6.2.2. Detecção dos Trifones.

A base de dados gerada para estes testes não apresenta todos os trifones possíveis. Considerando que a transcrição fonética das locuções foi realizada com 36 fones (35 fones independentes de contexto mais um, correspondente ao silêncio), teríamos, em termos grosseiros, 36^3 = 46656 trifones. A grande maioria deles não é observada nas locuções que constituem a base de dados. Desta forma, o dicionário de fones dependentes de contexto será limitado àqueles observados nas locuções de treinamento.

O procedimento adotado para a geração dos trifones é o seguinte: inicialmente são tomadas as transcrições fonéticas das locuções feitas com fones independentes de contexto. Para cada fone da transcrição são identificados o fone imediatamente anterior e o imediatamente posterior, gerando-se assim o trifone correspondente. Os trifones detectados são armazenados em uma lista. Deve-se observar que o fone inicial não tem o contexto à esquerda, e o fone final não tem o contexto à direita, mas como eles são sempre o fone correspondente ao silêncio, isto não chega a ser um problema. Aliás, não são gerados modelos trifones para o silêncio: ele é sempre considerado um fone independente de contexto.

São também atualizados os arquivos com as transcrições fonéticas para uma versão utilizando os trifones, e o arquivo de vocabulário utilizado no reconhecimento (ver seção 6.4).

É preciso ainda gerar modelos HMM para estas novas sub-unidades. Este programa se encarrega desta tarefa, atribuindo a cada modelo trifone os parâmetros dos modelos HMM dos fones independentes de contexto correspondentes. Estes modelos devem então ser retreinados utilizando o programa de treinamento descrito na seção 6.2.1. Um diagrama de blocos que mostra este procedimento é mostrado na Figura 12.

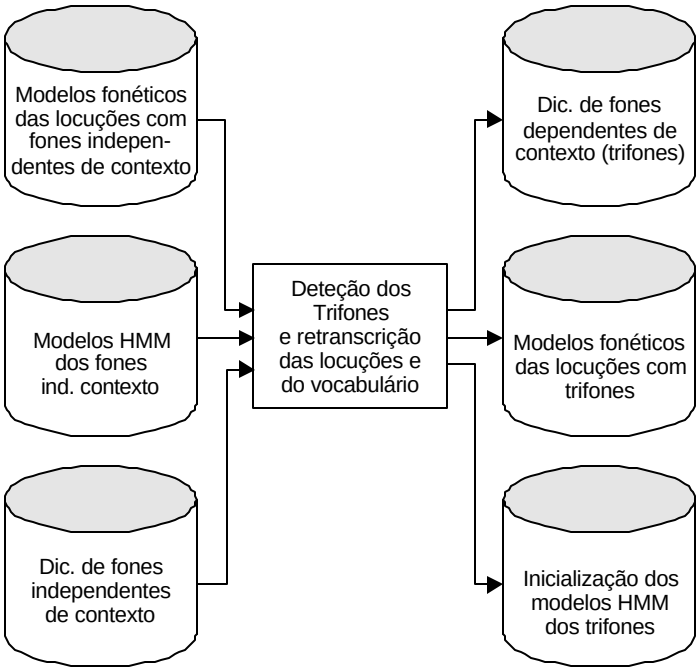


Figura 12: Diagrama de blocos para o programa de deteção de trifones.

A geração de todos os trifones contidos na base de dados criaria um conjunto de sub-unidades muito grande para a base de dados de treinamento. De fato, foram detectados 3655 trifones trifones somente no subconjunto de treinamento. Isto faz com que haja uma escassez muito grande de dados de treinamento para estimar de forma consistente todos os parâmetros envolvidos. A solução seria então agrupar os trifones em classes convenientemente definidas de modo a diminuir o número de sub-unidades acústicas sem perder a propriedade de consistência que estas possuem. LEE [30] propõe

um método baseado na medida de quantidade de informação da Teoria de Informação para determinar um número razoável de trifones baseado no tamanho da base de dados de treinamento. Neste trabalho foram testadas duas abordagens alternativas, baseadas em informações linguísticas.

Na primeira, a idéia é associar a cada um dos fones independentes de contexto uma etiqueta correspondente à sua classe fonética. As classes fonéticas utilizadas são: vogais, vogais nasais, plosivas, fricativas, laterais, vibrantes e nasais. O silêncio foi considerado como uma classe separada. Na Tabela 4 são listadas as classes fonéticas e os fones que as compõem:

Tabela 4: Classes fonéticas com seus respectivos fones.

Classes	Fones
Silêncio (s)	#
Vogais orais (v)	a, e, ɛ, i, j, o, ɔ, u
Vogais nasais (vn)	$\tilde{\alpha}$, $\tilde{e}$, $\tilde{i}$, $\tilde{o}$, $\tilde{u}$
Consoantes plosivas (p)	p, t, tʃ, k, b, d, dʒ, g
Consoantes fricativas (f)	f, s, ʃ, v, z, ʒ
Consoantes laterais (l)	l, ʎ
Consoantes nasais (n)	n, m, ɲ
Consoantes vibrantes (vb)	r, $\bar{r}$, R

Na segunda abordagem, as classes são definidas segundo uma classificação baseada na fonética acústica [9], onde é levada em conta a configuração do trato vocal. Neste caso, como existem muitas classes, algumas delas foram agrupadas para diminuir o seu número. Na Tabela 5 são listadas as classes e os respectivos fonemas para esta classificação.

Como uma ilustração do processo da substituição da transcrição fonética utilizando fones independentes de contexto para a transcrição utilizando trifones, considere uma locução correspondente à palavra ‘casa’. Utilizando fones independentes de contexto, poderíamos transcrevê-la como:

k a z a

Utilizando os trifones em sua forma tradicional, esta mesma locução teria a seguinte transcrição fonética:

s_k_a k_a_z a_z_a z_a_s

Usando os trifones gerados a partir da abordagem linguística da Tabela 4, teríamos o seguinte:

s_k_v p_a_f v_z_v f_a_s

Finalmente, utilizando os trifones baseados nos conceitos de fonética acústica, a transcrição da mesma locução teria a forma:

s_k_vm cp_a_cm vm_z_vm cm_a_s

No Capítulo 7 são realizados testes de reconhecimento utilizando fones independentes de contexto, trifones baseados nas classes fonéticas e trifones baseados na configuração do trato vocal. Também são feitos comentários a respeito das características e influência no desempenho do sistema no reconhecimento para cada um destes conjuntos de sub-unidades, bem como análises sobre o custo computacional e de armazenamento em memória.

Tabela 5: Classes fonéticas baseadas na posição do trato vocal e seus respectivos fones.

Classes	Fones
Silêncio (s)	#
Vogais anteriores (va)	i, j, e, ɛ, ẽ , ĩ
Vogais mediais (vm)	a, α̃
Vogais posteriores (vp)	o, ɔ, u, õ, ũ
Consoantes labiais (cl)	p, b, m, f, v
Consoantes mediais (cm)	t, tʃ, d, dʒ, n, s, z
Consoantes posteriores (cp)	k, g, ɲ, ʃ, ʒ
Laterais (l)	l, ʎ
Vibrantes (v)	R, r, r̄

6.2.3. Deleted Interpolation [15].

Como mencionado no Capítulo 1, os trifones, embora sejam unidades consistentes, não são treináveis, devido ao seu número muito elevado. Entretanto, podemos observar que os trifones correspondem a fones específicos e, deste modo, seus modelos podem ser interpolados com os dos fones independentes de contexto, que são melhor treinados, embora pouco consistentes. O *Deleted Interpolation* é um método para obter um modelo 'híbrido', que inclui automaticamente uma proporção adequada de cada um dos modelos originais [15].

O método pode ser resumido da seguinte maneira: seja *T* o conjunto de locuções de treinamento do sistema. Suponha que dividamos *T* em dois sub-conjuntos disjuntos, *T'* e *T"*.

Usamos *T'* para treinar os modelos dos fones independentes de contexto (M_f) e os modelos dos trifones (M_t). Depois, fazemos experimentos de reconhecimento de cada uma das sequências em *T"*, usando M_f e M_t em cada experimento. Em cada caso, um dos modelos irá produzir uma verossimilhança maior.

Seja ε_f a fração das sequências em *T"* para as quais M_f produziu verossimilhanças maiores. Se a_f e b_f são as matrizes com as probabilidades de transição e de emissão para M_f, e a_t e b_t as matrizes correspondentes para M_t, então o modelo interpolado terá matrizes:

$$a = \varepsilon_f a_f + (1 - \varepsilon_f) a_t \quad (42)$$

$$b = \varepsilon_f b_f + (1 - \varepsilon_f) b_t \quad (43)$$

Na verdade o termo *Deleted Interpolation* é geralmente usado para descrever um procedimento um pouco mais complicado do que o descrito acima. Neste caso, o conjunto de treinamento *T* é reparticionado iterativamente e o procedimento é repetido sobre cada partição. Existem muitas maneiras de construir as partições múltiplas. Por exemplo, T" pode ser composto pelos primeiros 10% de T na primeira iteração, pelos próximos 10% na segunda, e assim por diante.

Na abordagem por sub-unidades utilizada neste trabalho, o termo ε_f não pode ser obtido pela simples substituição da transcrição fonética em fones pela transcrição fonética em trifones de uma dada locução. Isto porque deve ser verificada a influência de cada trifone separadamente sobre o desempenho do sistema. Assim, o procedimento adotado foi o seguinte:

- Para cada locução do conjunto de validação *T"*:
- Inicialmente é formado o modelo de fones independentes de contexto da locução e calcula-se a verossimilhança correspondente
- Substitui-se apenas um fone independente de contexto pelo seu trifone correspondente e calcula-se a verossimilhança deste novo modelo.
- Repete-se este processo para todos os fones da locução.
- Para cada trifone calcula-se o percentual de vezes em que o uso do fone correspondente foi melhor que o uso do trifone. A partir do percentual de vezes em que a verossimilhança dos fones foi maior que a dos trifones, calcula-se o valor de ε_f.

Este procedimento produz valores ε_f diferentes para cada um dos trifones, e garante que a influência de cada um deles foi avaliada de forma particular.

Por escassez de dados de treinamento, optou-se por fazer a avaliação de ε_f com os mesmos dados utilizados para os testes. O sistema desenvolvido para implementar o algoritmo *Deleted Interpolation* é mostrado na Figura 13.

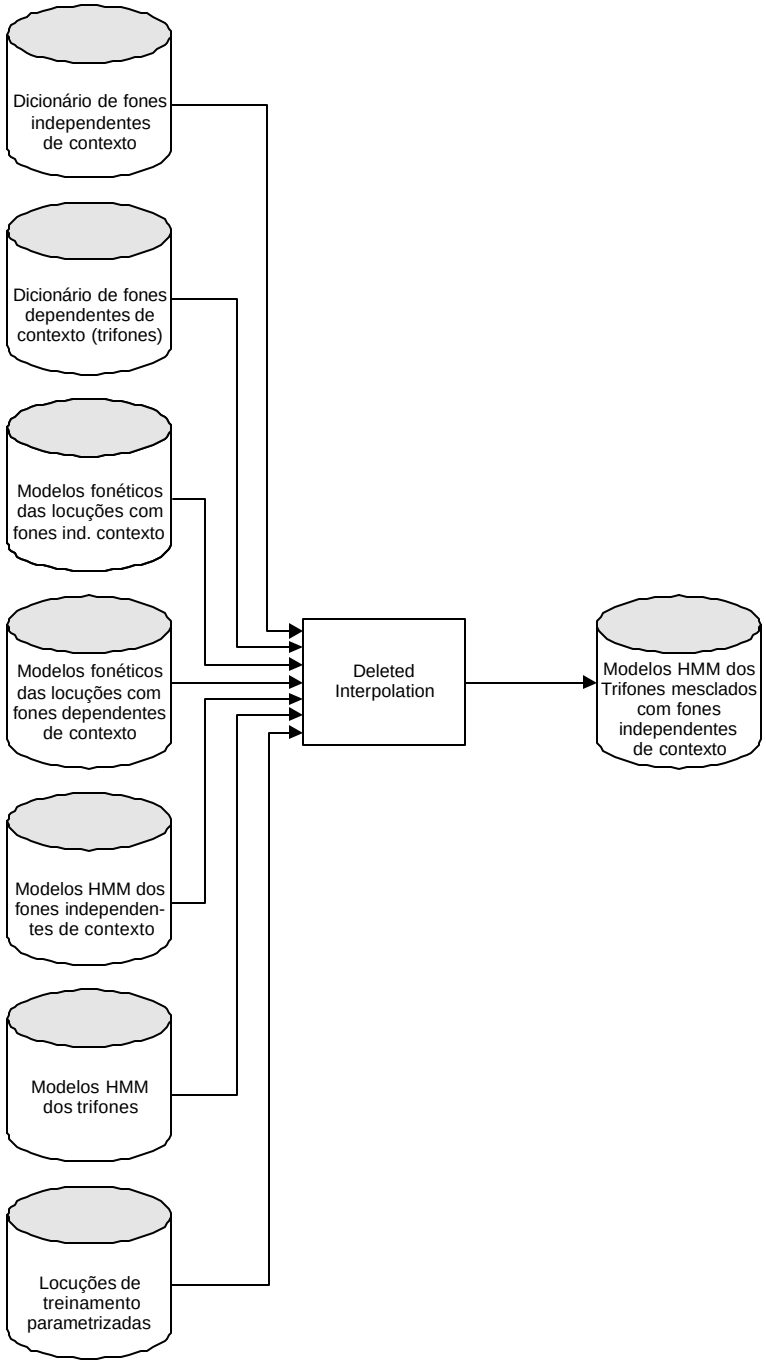


Figura 13: *Deleted Interpolation*.

6.3. Módulo de geração do modelo de linguagem.

Como discutido no Capítulo 2, o modelo de linguagem faz com que a perplexidade seja reduzida no processo de reconhecimento. Foi dito também que um modelo de linguagem robusto necessita de bases de dados extensas para um bom desempenho. A exemplo da base de dados de vozes, a construção de uma base de dados para treinamento do modelo de linguagem é bastante onerosa. Desta forma optou-se por utilizar apenas as frases utilizadas para gerar a base de dados como material de treinamento para o modelo de linguagem.

O procedimento adotado para a construção da gramática foi o seguinte:

1. Gera-se um arquivo texto com todas as frases a serem consideradas para o modelo de linguagem.
2. Realiza-se um levantamento das palavras que compõem as frases, bem como a frequência de ocorrência destas.
3. Realiza-se um levantamento das sequências de duas palavras que ocorrem nas frases, e também a frequência de ocorrência destas sequências.
4. As frequências de ocorrência de pares de palavras que podem ocorrer são transformadas em probabilidades através da expressão (4).

Os valores das probabilidades não foram utilizados neste trabalho. Como mencionado na seção 5.5, foi utilizada uma gramática de pares de palavras, que é um modelo simplificado da gramática bigrama. Neste caso, o sistema de reconhecimento verifica apenas se a probabilidade de uma sequência de palavras é nula ou não. O cálculo das frequências foi inserido neste programa visando trabalhos futuros.

Para um vocabulário de N palavras, seria necessário criar uma matriz N x N para armazenar todos os dados. Entretanto, como esta matriz é esparsa, não é necessário armazenar todos os elementos. Com isto foram armazenados, para cada posição não nula

desta matriz, os índices da primeira e segunda palavras e a probabilidade de uma seguir a outra.

6.4. Módulo de Reconhecimento.

O módulo de reconhecimento é o responsável pelo mapeamento dos parâmetros acústicos correspondentes à locução de entrada em sua transcrição ortográfica. Foram implementados dois algoritmos de busca para o reconhecimento de fala contínua: o *Level Building* e o *One Step*. Para melhorar o desempenho do sistema em termos de taxa de acertos foram incluídos o modelo de duração de palavras, citado na seção 2.4, e o modelo de linguagem bigrama, mostrado na seção 2.6.1. Também foram incorporadas estratégias para diminuição do tempo de processamento: o Viterbi *Beam Search* (ver seção 4.4.1) para o algoritmo *One Step* e um esquema de deteção automática do número de níveis de reconhecimento para o algoritmo *Level Building* (que será explicado na seção 6.4.2). Um diagrama de blocos para este sistema é mostrado na Figura 14.

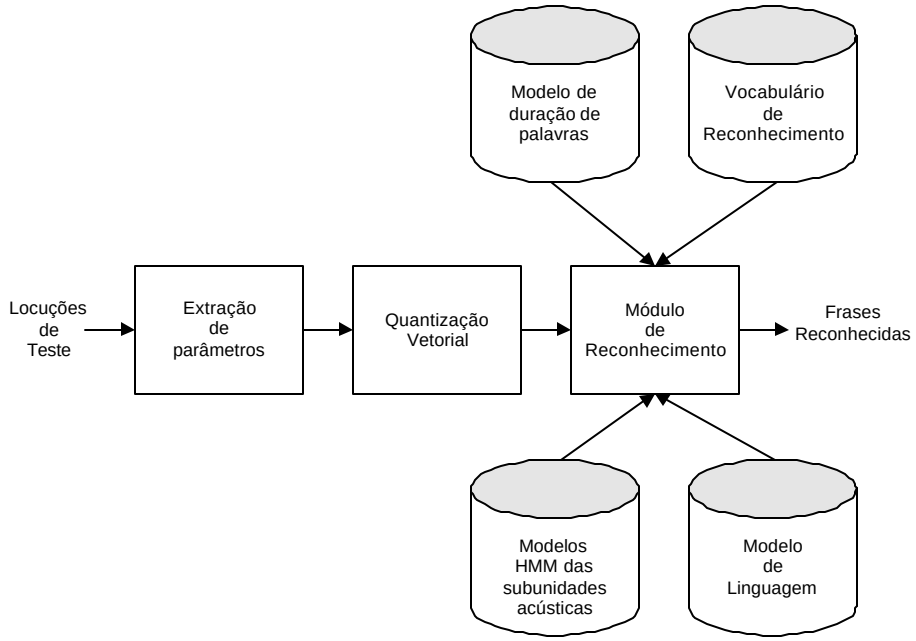


Figura 14: Diagrama de blocos do módulo de reconhecimento.

6.4.1. Construção do vocabulário de reconhecimento.

Observando-se a Figura 14, verifica-se que os dados necessário para o reconhecimento de uma dada locução são os parâmetros quantizados desta, os modelos HMM das sub-unidades acústicas e o vocabulário com o universo das palavras que podem ser reconhecidas. Os dois primeiros itens já foram abordados em seções anteriores, de modo que nesta seção será dada ênfase à construção do arquivo de vocabulário.

O vocabulário de um sistema de reconhecimento de fala é a unidade que define o universo de palavras que podem ser reconhecidas, ou seja, toda e qualquer locução será mapeada em uma sequência de palavras deste universo. Em termos gerais, quanto maior e mais abrangente o vocabulário, mais flexível é o sistema, embora o reconhecimento torne-se cada vez mais difícil à medida em que o vocabulário cresce. Com estas considerações em mente, neste trabalho o vocabulário foi definido a partir das frases que compõem a base de dados (200 frases foneticamente balanceadas).

Outra questão a ser abordada é a das diferenças de pronúncia. Dependendo do locutor, a mesma palavra pode ser pronunciada de várias maneiras diferentes. Como ressaltado no início desta tese, a variedade de pronúncias é muito grande e, se formos listar todas as variantes possíveis para todas as palavras do vocabulário, este se torna muito grande, aumentando muito a perplexidade no momento da busca. Se por um lado temos todas as variantes possíveis (ou que julgamos possíveis) para uma dada palavra, teoricamente temos um casamento mais preciso da locução com os modelos de palavras contidos no vocabulário. Entretanto, o aumento de modelos a serem comparados pode trazer mais dificuldades para o algoritmo de busca pelo aumento da perplexidade.

De modo a investigar estes efeitos, foram gerados dois arquivos de vocabulário para este sistema:

- Vocabulário 1: foi gerada apenas uma versão de cada palavra, ou seja, assumiu-se que todos os locutores pronunciaram as palavras da mesma maneira. Neste caso, o arquivo de vocabulário apresenta 694 palavras distintas.

- Vocabulário 2: neste, procurou-se cobrir a maior parte das pronúncias mais comuns, tentando prever alguns regionalismos e efeitos de coarticulação. Por exemplo, para a palavra ‘escola’ poderíamos ter as seguintes transcrições fonéticas: [e s k ɔ l a], [y s k ɔ l a], [y ʃ k ɔ l a], etc. Para evitar um aumento excessivo do vocabulário, foram consideradas no máximo seis variantes para cada palavra. Com estes cuidados, o vocabulário, que tem 694 palavras distintas, passou a ter 1633 palavras.

O arquivo de vocabulário é formado de duas partes: na primeira são listadas as sub-unidades fonéticas utilizadas para transcrever as palavras, e na segunda, a descrição das palavras, sua transcrição fonética e informação de duração das mesmas (média e variância das durações).

Como mencionado na seção 5.4, o modelo de duração foi levantado a partir das locuções de um único locutor. Nestas, algumas palavras ocorrem apenas uma vez, de modo que a variância da duração é nula. Para não prejudicar o reconhecimento com restrições tão rígidas, adotou-se para o desvio padrão da duração destas palavras um valor arbitrário correspondente a 1/3 do valor médio da duração.

Um exemplo do arquivo de vocabulário é mostrado na Figura 15. Nesta figura podemos observar o seguinte: inicialmente tem-se uma palavra chave (*fonemas), seguida de uma listagem das sub-unidades fonéticas utilizadas para a transcrição das palavras do vocabulário (uma sub-unidade por linha), e depois por outra palavra chave (*fim), que indica o final da listagem das sub-unidades. Em seguida, vem outra palavra chave (*vocab), que indica o início da listagem das palavras do vocabulário; para cada palavra (ou variante desta) tem-se uma linha com a seguinte estrutura: *transcrição gráfica / transcrição fonética / média da duração (ms) / desvio padrão da duração.*

```
*fonemas
#
a
an
...

v
x
z
*fim
*vocab
, / # / 0 / 0
a / a / 100.771084 / 705.260851
abertura / a b e rr t u r a / 490 / 0
abertura / a b e r t u r a / 490 / 0
...
voltar / v o u t a r / 600 / 0
voltar / v o u t a rr / 600 / 0
voltar / v o u t a / 600 / 0
...
*fim
```

Figura 15: Exemplo de arquivo de vocabulário

6.4.2. Detecção automática do número de níveis para o algoritmo *Level Building.*

Quando é utilizado o algoritmo *Level Building* para o reconhecimento, é necessário informar o número máximo *L* de palavras nas locuções. O algoritmo processa então os dados de entrada até o nível *L* e então decide quantas palavras existem na locução. O custo computacional para este algoritmo é $O(LMT)$, onde M é o número de palavras no vocabulário e T é o número de quadros da locução a ser reconhecida, e estas variáveis são fixas para um dado vocabulário e uma locução sendo reconhecida. Desta forma, quanto maior o valor de L, maior é o custo computacional e, consequentemente, o tempo de processamento. A diminuição de L diminui o tempo de processamento, mas pode causar erros de deleção em locuções longas. Como não se sabe a priori o número de palavras na locução, este valor deve ser relativamente alto. O inconveniente deste

procedimento é óbvio: o reconhecimento das locuções mais curtas levará um tempo desnecessariamente longo pois o sistema irá efetuar os cálculos até o nível L antes de fornecer uma resposta.

O ideal seria que o sistema pudesse determinar de forma automática o número de palavras na frase durante o processo de busca, e fazer os cálculos somente até o nível correspondente. Uma forma extremamente simples de se fazer isso é observar que, de uma maneira geral, a verossimilhança $P(O \mid \lambda)$ ou seja, a probabilidade do modelo λ gerar a sequência de observação O, tende a crescer com o número de níveis até atingir um pico, voltando a cair depois disso. Este comportamento de $P(O \mid \lambda)$ é ilustrado na Figura 16. Ainda, nos testes realizados notou-se que, geralmente, o ponto de máximo ocorre em um nível próximo ao número de palavras reconhecidas pelo sistema quando não é utilizado este procedimento.

Se o crescimento e decaimento de $P(O \mid \lambda)$ fossem sempre monotônicos, como mostrado na Figura 16, a decisão de parada poderia ser feita verificando apenas o valor de $P(O \mid \lambda)$ no nível anterior, e parando a busca no ponto de inflexão da curva. Entretanto, isto não é verdade, como pode ser visto na Figura 17. Neste caso, se fosse adotado o procedimento de verificar apenas a verossimilhança no nível anterior, o sistema reconheceria uma locução de 5 palavras, e não de 8, incorrendo em um erro de deleção bastante nocivo ao resultado, pois quase metade das palavras da locução não seriam detectadas.

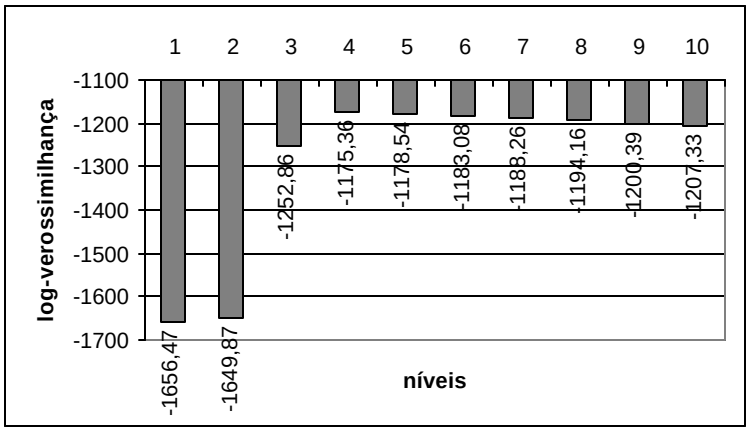


Figura 16: Variação de $P(O \mid l)$ com o número de níveis para uma locução de quatro palavras. Verifica-se um comportamento monotônico de crescimento e decaimento nos valores da log-verossimilhança com o número de níveis.

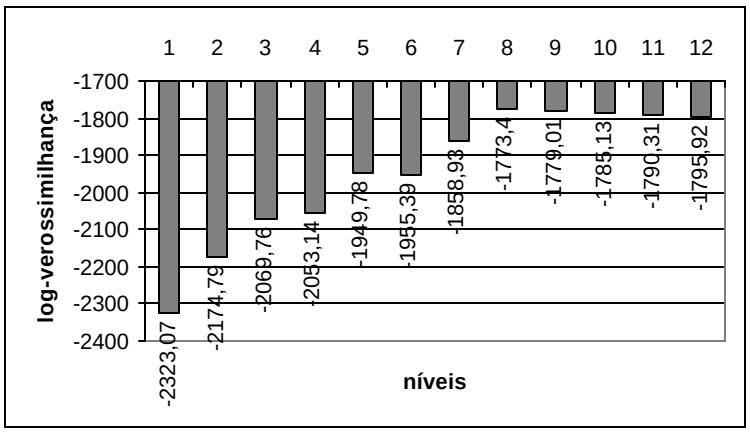


Figura 17: Variação de $P(O \mid \lambda)$ com o número de níveis para uma locução de oito palavras. Verifica-se um comportamento não monotônico de crescimento e decaimento nos valores da log-verossimilhança com o número de níveis.

Para resolver este problema, pensou-se inicialmente em estabelecer o critério de parada com base na derivada da curva log-verossimilhança versus número de níveis. O procedimento seria o seguinte: define-se a derivada da curva através da expressão

$$d = \frac{p_{atual} - p_{anterior}}{p_{atual}} \tag{44}$$

Pode-se ver da expressão acima que enquanto a log-verossimilhança cresce, o valor *d* é positivo. No decaimento, este valor torna-se negativo.

O procedimento para detecção automática do número de níveis utilizando esta medida é o seguinte: define-se um limiar para *d*, abaixo do qual o algoritmo para o processamento. Se este limiar for negativo, o sistema permite que haja não monotonicidades no comportamento da log-verossimilhança, desde que não sejam muito fortes.

A segunda alternativa encontrada foi estabelecer o seguinte critério de parada: o algoritmo para se a log-verossimilhança cair por *l* níveis consecutivos. Desta forma, o sistema pode atravessar as quebras no comportamento da log-verossimilhança que sejam menores do que *l*. Este procedimento pode ser visto como um detetor de tendência de queda no comportamento do valor da log-verossimilhança, uma vez que se este cai durante alguns níveis é muito provável que o máximo global já tenha sido atingido, e já esteja em regime de queda.

É importante ressaltar que estes procedimentos nem sempre diminuem o tempo de processamento. Se for utilizado o procedimento de fixar um número de níveis de busca e este for o número de palavras da locução, a quantidade de cálculos efetuados pelo sistema será o estritamente necessário, enquanto que no modo de detecção automática, pode ocorrer de o sistema avançar mais alguns níveis.

De uma forma geral, entretanto, o procedimento proposto gera uma economia de esforço computacional, pois o sistema irá sempre realizar os cálculos com um número de níveis próximo ao de palavras na locução. Ainda, a economia será maior nas locuções mais curtas.

Foram realizados alguns testes para verificar o quanto se pode ganhar em tempo de processamento, e os resultados podem ser vistos no capítulo a seguir.

7. Testes e análise dos resultados.

7.1. Introdução.

Neste capítulo são apresentados os resultados de avaliação do sistema desenvolvido utilizando a base de dados descrita no Capítulo 3. Foram realizados vários testes, e feitas as seguintes análises:

- desempenho do sistema utilizando fones independentes de contexto.
- desempenho do sistema utilizando fones dependentes de contexto baseados nas classes fonéticas.
- desempenho do sistema utilizando fones dependentes de contexto baseados na posição do trato vocal.
- avaliação dos procedimentos para diminuição do tempo de processamento.
- influência da dependência do locutor com testes dependentes de locutor, independentes de locutor e dependentes de sexo.
- influência da transcrição fonética das locuções de treinamento no desempenho do sistema.
- influência do número de versões de cada palavra no arquivo de vocabulário.

Antes dos testes propriamente ditos foram realizados alguns testes preliminares para a determinação do conjunto de sub-unidades fonéticas a serem utilizadas neste trabalho. A motivação e o procedimento são descritos na seção seguinte.

Para todos os testes foram utilizados parâmetros mel-cepstrais, delta mel-cepstrais e delta-delta mel-cepstrais, quantizados separadamente, modelo de duração de palavras e gramática de pares de palavras. Os parâmetros log-energia normalizada, bem como suas derivadas não foram utilizados pois foi detectada uma piora no desempenho do sistema quando adotados.

Os testes são descritos nas seções 7.2 a 7.9. Na seção 7.10 são feitas análises e comentários sobre os resultados obtidos.

7.2. Determinação do conjunto de sub-unidades fonéticas.

No português falado no Brasil existem 39 fones distintos [9]. Na Tabela 6 são listados os fones utilizados na variante brasileira da língua portuguesa.

Como dito anteriormente, alguns deles são bastante próximos e seria interessante agrupá-los, ou seja, considerá-los como sendo o mesmo fonema, pois isto diminuiria o número de sub-unidades fonéticas, fazendo com que houvesse mais exemplos de treinamento para cada uma. Entretanto, este agrupamento deve ser feito de forma a não juntar fones com características diferentes, o que pode fazer com que as sub-unidades resultantes se tornem inconsistentes. As fusões testadas foram:

- [i] e [j]
- [u] e [w]
- [R] e [$\overline{R}$]
- [a] e [α]

- *[e] e [ə][4]

Tabela 6: Lista dos fones presentes no português falado no Brasil.

Vogais orais		Consoantes orais	
[i]	l*i*vro	[p]	*p*á
[e]	P*e*dro	[b]	*b*ata
[ɛ]	t*e*rra	[t]	*t*arde
[a]	p*a*to	[d]	*d*ado
[ɑ]	m*a*no	[k]	*c*ão
[ɔ]	g*o*la	[g]	*g*ato
[o]	p*o*ço	[s]	*s*ábado
[u]	p*u*lar	[z]	casa
[ə]	s*e*car	[ʃ]	*ch*ão
		[ʒ]	*j*ardim
		[f]	*f*ado
Vogais nasais		[v]	*v*aca
		[l]	*l*ado
[ĩ]	p*in*to	[ʎ]	fi*lh*o
[ẽ]	d*en*te	[r]	por*t*a
[α̃]	c*an*to	[r̄]	car*r*o
[õ]	p*on*te	[R]	por*t*a (velar)
[ũ]	f*un*do	[R̄]	car*r*o (velar)
		[tʃ]	*t*ia
		[dʒ]	*D*ia
Semivogais		Consoantes nasais	
[j]	pa*i*	[m]	*m*ãe
[w]	pa*u*	[n]	*n*ada
		[ʎ]	pi*nh*o

O procedimento utilizado para verificar se uma fusão deveria ou não ser adotada foi o seguinte:

[4] Na verdade esta fusão foi feita já na transcrição fonética original, e não foi nem testada nesta etapa.

- Inicialmente foram gerados e treinados os modelos HMM de todos os 39 fones listados na Tabela 6.
- Com esses modelos calculou-se a probabilidade média dos modelos HMM das locuções de treinamento gerarem as sequências de observação correspondentes. Esta probabilidade é tomada então como referência.
- Para cada uma das fusões propostas acima, foram criados e testados os modelos HMM correspondentes e calculada novamente a probabilidade de os modelos gerarem as sequências de observação. Se esta probabilidade fosse maior que a de referência, a fusão era adotada. Na Tabela 7 tem-se os resultados destes testes.

Tabela 7: Resultados dos testes realizados para fusão de fones independentes de contexto.

Testes	log (P(O|λ))
todos os fones independentes de contexto (referência)	-1693.13
a) juntando [i] e [j]	-1693.65
b) juntando [u] e [w]	-1692.96
c) juntando [$\overline{R}$] e [R]	-1693.21
d) juntando [a] e [α]	-1693.05

Com este procedimento, foram adotadas todas as fusões testadas, exceto a primeira, chegando-se então às unidades listadas na Tabela 3. A única fusão adotada que não resultou em uma diminuição da verossimilhança média foi a do teste c) (juntando [$\overline{R}$] e [R]). Mesmo assim ela foi adotada uma vez que, na transcrição fonética, nem sempre era possível ter certeza da ocorrência de um ou outro, o que poderia causar erros.

7.3. Definição dos subconjuntos de teste e treinamento.

A base de dados coletada é formada por 40 locutores, sendo 20 do sexo masculino e 20 do sexo feminino. Como mencionado na seção 3.3.2, os locutores foram

separados em 5 grupos, onde cada grupo pronunciou 4 das 20 listas. Deste modo, temos 4 locutores de cada sexo em cada grupo.

Para a formação do subconjunto de teste foram escolhidos de cada grupo, e de forma aleatória, um locutor do sexo masculino e um do sexo feminino, resultando no total 5 locutores femininos e 5 masculinos. Os demais locutores, 15 masculinos e 15 femininos, formam o subconjunto de treinamento. Nos testes com dependência de sexo, os locutores de treinamento e teste são extraídos dos subconjuntos anteriores, resultando em 5 locutores de teste e 15 de treinamento. A divisão dos locutores é ilustrada na Figura 18.

Para os testes com dependência de locutor, uma única pessoa do sexo masculino pronunciou todas as frases 3 vezes. Duas repetições formam o subconjunto de treinamento e a terceira, o subconjunto de testes.

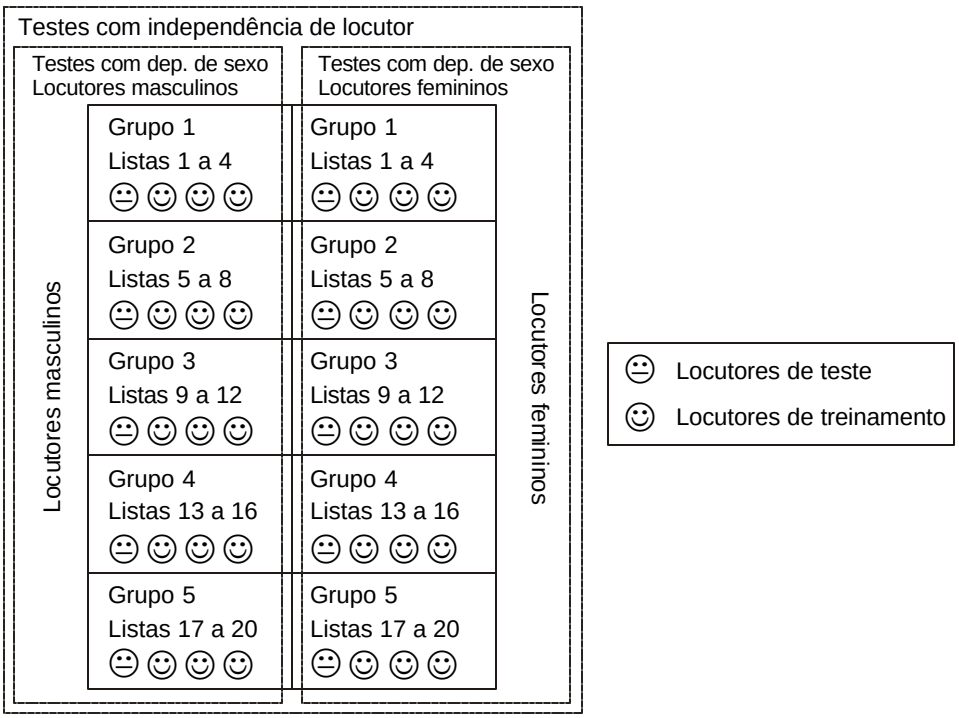


Figura 18: Divisão dos locutores em conjuntos de treinamento e teste.

7.4. Testes com fones independentes de contexto

Os primeiros testes foram realizados utilizando os fones independentes de contexto, listados na Tabela 3. Estes testes têm por finalidade estabelecer um desempenho de referência a partir do qual será analisada a influência dos fones dependentes de contexto no desempenho do sistema desenvolvido.

Nesta etapa foi utilizado o algoritmo *Level Building* com 15 níveis para todas as locuções. A escolha deste número de níveis está relacionada às frases da base de dados. A frase mais longa tem 11 palavras, e contando os silêncios inicial e final, temos 13 palavras. Com 15 níveis é possível reconhecer todas as frases, e ainda verificar se ocorrem erros de inclusão, mesmo nas frases mais longas.

Foram realizados 4 testes, variando-se os locutores envolvidos:

- teste com independência de locutor (todos os 10 locutores de teste).
- teste com dependência de sexo para os locutores masculinos (5 locutores de teste do sexo masculino).
- teste com dependência de sexo para os locutores femininos (5 locutores de teste do sexo feminino).
- teste com dependência do locutor (1 locutor do sexo masculino).

Os resultados destes testes são mostrados resumidamente na Tabela 8.

Tabela 8**:** taxa de erro de palavra (%) para os testes com fones independentes de contexto

Locutores	Deleção	Substituição	Inserção	total
Independente	4,72	14,00	2,59	21,31
Masculinos	4,72	11,49	2,74	18,95
Femininos	4,34	14,84	2,51	21,69
Dependente	2,21	6,62	1,67	10,50

7.5. Testes com trifones.

Uma vez estabelecida uma referência para a taxa de acertos do sistema, foram realizados testes para verificar a influência dos fones dependentes de contexto no seu desempenho. Como mencionado na seção 6.2.2, foram testados dois conjuntos de fones dependentes de contexto: um baseado nas classes fonéticas e outro baseado na configuração do trato vocal.

O levantamento dos trifones é feito através das transcrições fonéticas das locuções de treinamento, de modo que para cada teste (dependente de locutor, independente, etc.) temos um número diferente destes, devido à variação na pronúncia dos locutores envolvidos.

Um comentário deve ser feito acerca do arquivo de vocabulário do sistema: as palavras contidas neste vocabulário são as mesmas da base de dados. Entretanto, as transcrições das mesmas foram feitas tentando prever como as pessoas poderiam pronunciá-las, o que nem sempre ocorre. Desta forma, alguns trifones podem não constar da lista de sub-unidades treinadas. Neste caso, estes trifones foram substituídos pelos fones independentes de contexto correspondentes. Da mesma maneira, o primeiro e o último trifone da palavra foram substituídos pelos respectivos fones independentes de contexto pois, no caso do primeiro fone, não se conhece o contexto à esquerda e, no caso do último, não se conhece o contexto à direita.

Como no caso anterior, foi utilizado o algoritmo *Level Building*, com 15 níveis.

7.5.1. Trifones baseados nas classes fonéticas.

Utilizando as classes fonéticas listadas na Tabela 4 chegou-se aos seguintes números de fones dependentes de contexto:

Tabela 9: número de trifones baseados nas classes fonéticas gerados a partir do subconjunto de locuções de treinamento.

Locutores	Número de trifones
Independente	717
Masculinos	674
Femininos	678
Dependente	586

Os resultados dos testes realizados com estes conjuntos de sub-unidades fonéticas podem ser vistos na Tabela 10:

Tabela 10**:** taxa de erro de palavra (%) para os testes com trifones baseados nas classes fonéticas.

Locutores	Deleção	Substituição	Inserção	total
Independente	4,53	13,62	2,36	20,51
Masculinos	3,81	11,57	2,21	17,59
Femininos	3,27	15,37	3,88	22,52
Dependente	1,98	6,70	1,97	10,65

7.5.2. Trifones baseados na configuração do trato vocal.

Para os testes com trifones baseados na configuração do trato vocal, foram utilizadas as classes definidas na Tabela 5. O número de trifones gerados é mostrado na Tabela 11, e os resultados dos testes são apresentados na Tabela 12.

Tabela 11: número de trifones baseados na configuração do trato vocal gerados a partir do subconjunto de locuções de treinamento.

Locutores	Número de trifones
Independente	1018
Masculinos	959
Femininos	956
Dependente	829

Tabela 12: taxa de erro de palavra (%) para os testes com trifones baseados na configuração do trato vocal.

Locutores	Deleção	Substituição	Inserção	Total
Independente	4,45	13,24	2,47	20,16
Masculinos	3,96	11,57	2,36	17,89
Femininos	4,34	14,38	2,44	21,16
Dependente	1,90	6,01	1,90	9,81

7.6. Avaliação dos procedimentos para diminuição do tempo de processamento.

Os testes para avaliação dos procedimentos para diminuição do tempo de processamento na etapa de busca (parada automática para o *Level Building*, e Viterbi *Beam Search* para o *One Step*) foram realizados apenas utilizando a base dependente de locutor, e os trifones gerados a partir da posição do trato vocal. Os tempos de reconhecimento foram obtidos com base em um microcomputador PC com processador AMD-K6 350 MHz e 64 MB de memória RAM.

O primeiro passo foi estabelecer um tempo padrão de referência em relação ao qual seriam comparados os resultados. Foi realizado um teste de reconhecimento utilizando 15 níveis de busca para ambos os algoritmos de busca (*Level Building* e *One Step*), e o tempo médio de reconhecimento foi adotado como o padrão de referência.

7.6.1. Level Building.

Para o *Level Building* foram testadas as duas idéias para a redução no tempo de processamento na etapa de busca: parada pela derivada da curva de evolução da curva de log-verossimilhança com o número de níveis e parada pela contagem de níveis em que ocorre queda na log-verossimilhança.

Para o primeiro procedimento foram dois testes, variando-se o limiar de parada, e os resultados podem ser vistos na Tabela 13. Para o segundo procedimento também foram feitos dois testes, variando-se o número de níveis em que se observa queda no valor da log-verossimilhança. Os resultados para este segundo procedimento são mostrados na Tabela 14.

Tabela 13: Comparação do tempo médio de reconhecimento e taxa de erro de palavra para o procedimento de detecção automática do número de níveis baseado na derivada da curva de evolução da log-verossimilhança com o número de níveis.

limiar	tempo (min.)	erro de palavra (%)			
		D	S	I	total
15 níveis	05:58	1,90	6,01	1,90	9,81
d = -0,0040	05:49	1,90	6,01	1,90	9,81
d = -0,0035	05:42	2,21	6,01	1,90	10,12
d = -0,0030	05:19	4,19	5,86	1,52	11,57

Tabela 14: Comparação do tempo médio de reconhecimento e taxa de erro de palavra para o procedimento de detecção automática do número de níveis de acordo com a contagem do número de níveis em que a verossimilhança cai.

critério de parada	tempo (min.)	erro de palavra (%)			
		D	S	I	total
15 níveis	05:58	1,90	6,01	1,90	9,81
$l = 2$	05:04	1,90	5,78	1,52	9,20
$l = 1$	04:33	3,27	5,33	1,29	9,89

7.6.2. One Step.

O procedimento *Beam Search* foi testado variando-se o limiar de poda Δ e verificando-se o compromisso entre a taxa de erro de palavra e o tempo de processamento. Os resultados referem-se a testes realizados com o algoritmo *One Step* com 15 níveis.

Tabela 15: Comparação do tempo médio de reconhecimento e taxa de erro de palavra para vários valores do limiar de poda no algoritmo Viterbi *Beam Search*.

Limiar de poda	tempo (min)	erro de palavra (%)			
		D	S	I	total
$\Delta = 15$	02:48	2,21	12,25	2,74	17,20
$\Delta = 20$	02:56	2,05	7,61	2,21	11,87
$\Delta = 25$	03:20	1,83	6,54	1,75	10,12
$\Delta = 30$	03:40	1,60	6,46	1,75	9,81
$\Delta = 0$ (sem *Beam Search*)	06:52	1,90	6,01	1,90	9,81

7.7. Verificação da influência da transcrição fonética das locuções de treinamento no desempenho do sistema.

A confecção de uma base de dados compreende dois processos: a gravação das locuções e a transcrição fonética das mesmas. Esta última tarefa em especial é bastante penosa e demorada, pois é necessário ouvir com atenção as locuções, e com a ajuda de programas de visualização gráfica da forma de onda e do espectro do sinal, estabelecer exatamente o que foi pronunciado. Quando se realiza esta tarefa para milhares de locutores, cada qual pronunciando centenas de frases, verifica-se que o trabalho e tempo necessários são bastante grandes.

Poderia-se aliviar a carga de trabalho necessária para a confecção da base de dados se no processo de transcrição fonética fosse adotada uma transcrição padrão para todas as locuções, isto é, dada uma frase a ser pronunciada por vários locutores, faz-se uma transcrição fonética para um dos locutores, e esta é adotada para todas as locuções daquela mesma frase.

Espera-se que com este procedimento, o desempenho do sistema caia, pois um fonema poderia estar sendo treinado com a locução de outro. Entretanto a questão é: quanto? Talvez a queda verificada no desempenho não seja tão grande, e com essa

facilidade talvez seja possível construir bases de dados maiores, o que significa mais exemplos de treinamento e, consequentemente, sub-unidades fonéticas mais bem treinadas. Este compromisso pode fazer com que, mesmo que as transcrições fonéticas padronizadas atrapalhem o treinamento, o maior número de exemplos de treinamento acabe por compensar a queda no desempenho provocada por este procedimento.

Para testar esta idéia, o sistema foi treinado tomando-se as transcrições fonéticas das locuções dos testes dependentes de locutor e associando-as às locuções dos 30 locutores de treinamento da base de dados independente de locutor. Os testes foram realizados utilizando os trifones gerados a partir da configuração do trato vocal e comparados com os resultados obtidos na seção 7.5.2. O algoritmo de busca foi o *Level Building*, com 15 níveis, e os resultados dos testes são mostrados na Tabela 16.

Tabela 16: Desempenho do sistema em função das transcrições fonéticas das locuções de treinamento.

Testes	Erros(%)			
	Deleção	Subst.	Inserção	Total
transcrição original	4,45	13,24	2,47	20,16
transcrição padronizada	4,11	13,24	2,97	20,32

7.8. Influência do número de versões de cada palavra no arquivo de vocabulário.

Com os resultados obtidos nos testes da seção anterior, pôde-se verificar que uma transcrição fonética padronizada para todos os locutores não degrada de forma apreciável o desempenho do sistema. A partir deste resultado, foi investigada a influência que teria a mesma idéia quando aplicada ao arquivo de vocabulário.

A vantagem deste procedimento é a diminuição do espaço de busca: ao invés de o sistema testar, a cada nível, 1633 palavras, testaria apenas 694, o que corresponde a uma redução significativa no número de cálculos a serem realizados. Com isso, pode-se ganhar bastante em termos de tempo de processamento.

Comentou-se na seção 6.4.1 que foram construídos dois arquivos de vocabulário: um com várias versões de cada palavra, correspondendo às várias formas de locução, resultantes das diferenças de sotaque, coarticulações, etc., e outro, com apenas uma versão para cada palavra. Este segundo arquivo de vocabulário foi derivado do primeiro, escolhendo-se apenas uma variante e excluindo as demais (no caso das palavras com mais de uma versão). O critério adotado para a escolha da versão de cada palavra foi: a variante a ser selecionada é a que ocorreu com mais frequência nas locuções da base de dados.

Foram feitos testes com o arquivo de vocabulário simplificado, utlizando dois conjuntos de subunidades: os fones independentes de contexto e os trifones baseados na configuração do trato vocal. Não foram realizados testes com os trifones baseados nas classes fonéticas porque estas subunidades não apresentaram bons resultados nos testes anteriores.

Na Tabela 17 são apresentados os resultados dos testes realizados com o vocabulário simplificado, utilizando fones independentes de contexto. Na Tabela 18, os resultados para testes com trifones baseados na configuração do trato vocal. Finalmente, na Tabela 19, tem-se um quadro comparativo dos tempos de processamento utlizando os dois arquivos de vocabulário. Para todos estes testes foi utilizado o algoritmo *Level Building* com 15 níveis, modelo de duração de palavras e modelo de linguagem de pares de palavras.

Tabela 17: Resultados dos testes com vocabulário simplificado (apenas 1 versão de cada palavra), utilizando fones independentes de contexto.

Testes	Deleção	Substituição	Inserção	Total
Independente	5,97	13,05	1,79	20,81
Masculinos	4,34	11,64	2,28	18,26
Femininos	4,49	13,77	2,44	20,70
Dependente	2,05	6,09	1,83	9,97

Tabela 18: Resultados dos testes com vocabulário simplificado (apenas 1 versão de cada palavra), utilizando trifones baseados na configuração do trato vocal.

Testes	Deleção	Substituição	Inserção	Total
Independente	4,76	12,06	1,94	18,76
Masculinos	3,35	10,05	2,21	15,61
Femininos	4,49	13,24	1,90	19,63
Dependente	1,75	5,40	1,83	8,98

Tabela 19: tempo médio de reconhecimento para os testes com os dois arquivos de vocabulário.

Testes	Tempo médio de reconhecimento			
	Vocabulário completo		Vocabulário reduzido	
	fones	trifones	fones	trifones
Independente	05:21	05:17	02:05	02:10
Masculino	05:00	04:57	02:02	02:04
Feminino	05:35	05:40	02:22	02:24
Dependente	06:13	05:58	02:29	02:32

7.9. Estabelecimento do desempenho final do sistema.

Foi realizada uma rodada final de testes para estabelecer qual seria o desempenho final do sistema, utilizando as técnicas que proporcionaram os maiores ganhos ao sistema, tanto em termos de taxa de acerto como de tempo de processamento. Analisando os resultados de todos os testes anteriores, chega-se à conclusão que a configuração ideal deste sistema seria a seguinte (para a tarefa específica deste trabalho):

- Algoritmo de busca: *One Step* com Viterbi *Beam Search* (limiar de poda $\Delta = 30$).
- Subunidades fonéticas: trifones baseados na configuração do trato vocal.
- Arquivo de vocabulário: apenas uma versão de cada palavra.

Os demais parâmetros do sistema (parâmetros das locuções, quantização vetorial, modelo de duração de palavras e modelo de linguagem) permanecem os mesmos dos testes anteriores. Os resultados dos testes realizados com esta configuração são mostrados na Tabela 20.

Tabela 20: Resultados dos testes de avaliação do desempenho final do sistema.

Testes	% erros				tempo médio
	D	S	I	total	
Independente	4,38	12,25	2,36	18,99	01:17
Masculino	3,65	9,89	1,83	15,37	01:12
Feminino	4,49	13,09	2,05	19,63	01:24
Dependente	1,60	5,63	1,90	9,13	01:26

7.10.Análise dos resultados.

Neste capítulo foram apresentados os testes de avaliação do sistema implementado, utilizando a base de dados descrita no Capítulo 3. Foram avaliados:

- o desempenho do sistema utilizando fones independentes de contexto e a influência do modo de operação do sistema (dependente de locutor, dependente de sexo e independente de locutor) na taxa de acertos;
- a influência de dois tipos de fones dependentes de contexto na taxa de acertos;
- a influência dos procedimentos de diminuição dos cálculos necessários na etapa de busca no tempo de reconhecimento;
- a influência da transcrição fonética das frases de treinamento no desempenho do sistema.
- a influência do número de versões de cada palavra no arquivo de vocabulário.
- o desempenho final do sistema.

A seguir, cada um destes itens será analisado com maiores detalhes.

7.10.1. Desempenho do sistema utilizando fones independentes de contexto e influência do modo de operação na taxa de acertos de palavra.

Estes testes iniciais serviram para estabelecer uma base de comparação para as melhorias implementadas no sistema. Pode-se verificar da Tabela 3 que as sub-unidades apresentam um número de exemplos de treinamento razoável, sendo que a sub-unidade com menos exemplos é o [ʃ] com 132 ocorrências.

As taxas de acerto são razoavelmente boas, com um índice de aproximadamente 80% de acerto de palavra para o caso independente de locutor, chegando a quase 90% no caso dependente de locutor.

Esperava-se que para os testes com dependência de sexo, os resultados fossem ficar entre estes dois extremos, o que realmente aconteceu com os testes utilizando os locutores masculinos. Entretanto, para os testes com locutores femininos, a taxa de acertos ficou abaixo dos testes realizados com independência de locutor.

Uma possível causa para este resultado é a presença de um locutor feminino para o qual o sistema apresentou um resultado muito ruim. Para investigar este fato, levantou-se inicialmente o número de erros de palavra cometidos pelo sistema para cada um dos locutores de teste. Este levantamento inicial foi feito para os testes com independência de locutor, e é mostrado na Figura 19.

Analisando a Figura 19, verifica-se que existem dois locutores para os quais o desempenho do sistema foi relativamente pior do que para os demais: f20 (feminino) e m23 (masculino). Nos testes com dependência de sexo, este comportamento se repetiu, embora menos acentuadamente para os locutores masculinos, como mostrado na Figura 20.

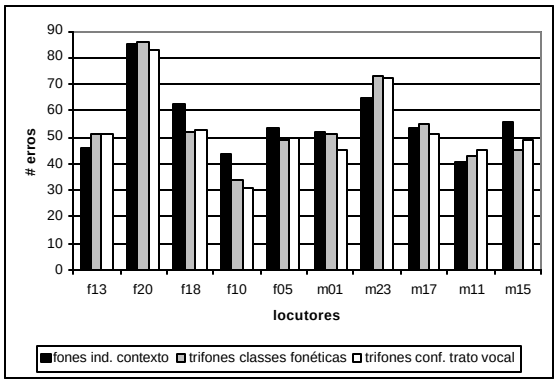
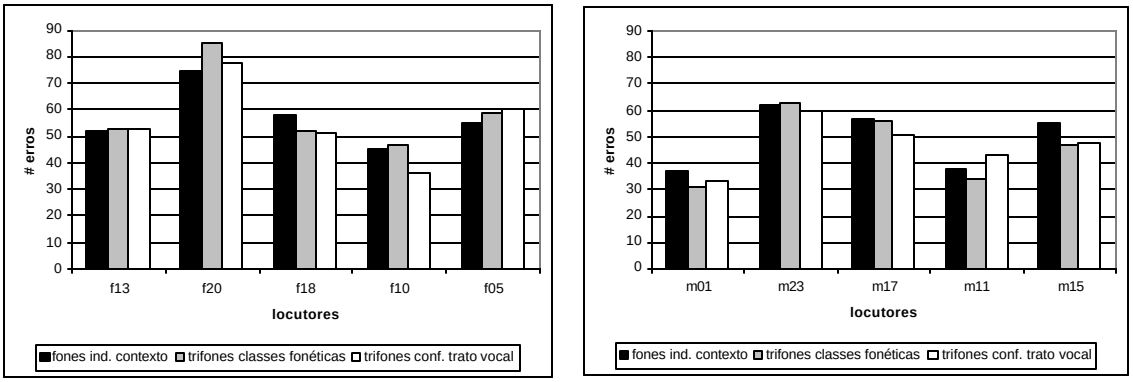


Figura 19: número de erros cometidos pelo sistema para cada locutor, para os testes com independência de locutor.



a) b)

Figura 20: número de erros cometidos pelo sistema para cada locutor, para os testes com dependência de sexo. a) locutores femininos e b) locutores masculinos.

Coincidentemente, os dois locutores pronunciaram o mesmo conjunto de frases (listas 5 a 8). Este subconjunto de frases poderia apresentar maiores dificuldades para o reconhecimento, e portanto o problema não estaria nos locutores. Para investigar este fato, levantou-se o histograma de erros para os testes com dependência de locutor, que é mostrado na Figura 21. Uma análise desta figura derruba a hipótese de que o subconunto de frases formado pelas listas 5 a 8 apresenta maiores dificuldades para o reconhecimento. Os piores desempenhos foram observados nos subconjuntos formados pelas listas 17 a 20 no caso dos fones independentes de contexto, 9 a 12 para os trifones baseados nas classes fonéticas, e 13 a 16 para os trifones baseados na configuração do trato vocal.

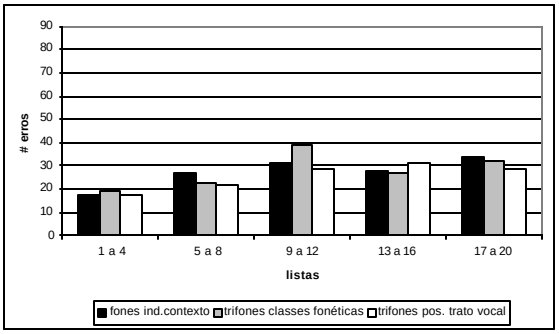


Figura 21: número de erros para cada subconjunto de frases nos testes com dependência de locutor.

Estes resultados parecem indicar que, de fato, a presença de um locutor feminino (f20) para o qual o desempenho do sistema foi bastante ruim, está polarizando os resultados.

Em relação ao modelo de linguagem, este mostrou ser bastante eficaz no direcionamento do processo de busca. Entretanto, alguns erros não puderam ser evitados:

- o sistema não é capaz de discernir palavras que tenham a mesma transcrição fonética. Deste modo, existem muitos erros de substituição entre as palavras a, à e há, por exemplo.
- podem também ocorrer erros de deleção de palavras curtas (geralmente artigos) quando precedem palavras que se iniciam com o mesmo fonema. Por exemplo, as frases ‘a atriz’ e ‘o ônibus’ são geralmente reconhecidas como ‘atriz’ e ‘ônibus’, com deleção dos artigos.

Isto mostra que a decodificação acústica está sendo bem feita, visto que no primeiro caso, todas as palavras são pronunciadas da mesma maneira e, no segundo caso, os locutores não se dão ao trabalho de pronunciar separadamente o artigo. Estes erros poderiam ser corrigidos com o uso de gramáticas sensíveis a contexto ou com parsers. As gramáticas sensíveis a contexto verificam a possibilidade de sequências de

palavras de acordo com a função sintática das palavras dentro da frase, e os parsers, além disso, procuram verificar o significado semântico da frase reconhecida.

7.10.2. Influência dos fones dependentes de contexto no desempenho do sistema.

A aglutinação dos fones em classes fonéticas mostrou ser útil na redução do número total de trifones gerados. Entretanto, esta aglutinação deve ser consistente para que os fones pertencentes a uma mesma classe tenham influências próximas nos fones adjacentes. Uma olhada nas tabelas 8, 10 e 12 mostra que, sob este ponto de vista, os trifones baseados nas classes fonética não são consistentes, chegando a atrapalhar o desempenho do sistema nos testes com locutores femininos e dependente de locutor.

Para o caso dos trifones gerados a partir da configuração do trato vocal, notou-se uma melhora pequena, mas consistente em todos os resultados. Esta melhora é um pouco mascarada pelo modelo de linguagem que, por ser bastante restritivo, pode fazer com que o desempenho dos testes com fones independentes de contexto tenham um resultado acima do que se poderia esperar. Talvez a utilização de uma gramática gerada a partir de mais exemplos possa dar uma idéia melhor do ganho que se obtém com os modelos trifones.

Outro fator que pode estar prejudicando o desempenho dos trifones é o reduzido número de exemplos de treinamento para cada uma destas sub-unidades. Na Figura 22 são mostrados gráficos em forma de histogramas onde são contados o número de sub-unidades fonéticas com menos de 10 exemplos de treinamento, o número de sub-unidades com menos de 20 exemplos, e assim por diante, para todos os testes realizados. Pode-se notar que a grande maioria das sub-unidades tem menos de 20 exemplos de treinamento, enquanto que para os fones independentes de contexto, o número mínimo de exemplos de treinamento foi 132 para o fone [ʃ]. Este fato é corroborado no processo de interpolação dos trifones com os fones independentes de contexto através do algoritmo *Deleted Interpolation*: na grande maioria dos casos, os fones independentes de

contexto apresentaram uma verossimilhança maior do que os trifones, indicando claramente que estes modelos estão mal treinados.

Neste trabalho, foram gerados 1018 trifones baseados na configuração do trato vocal, para o caso independente do locutor. Sistemas comerciais de fala contínua trabalham com pelo menos o dobro de trifones. Desta forma, para se conseguir unidades razoavelmente bem treinadas torna-se necessária uma base de dados muitíssimo maior do que a que foi utilizada neste trabalho. Outra alternativa seria gerar um conjunto menor de sub-unidades que pudesse ser treinada com menos exemplos.

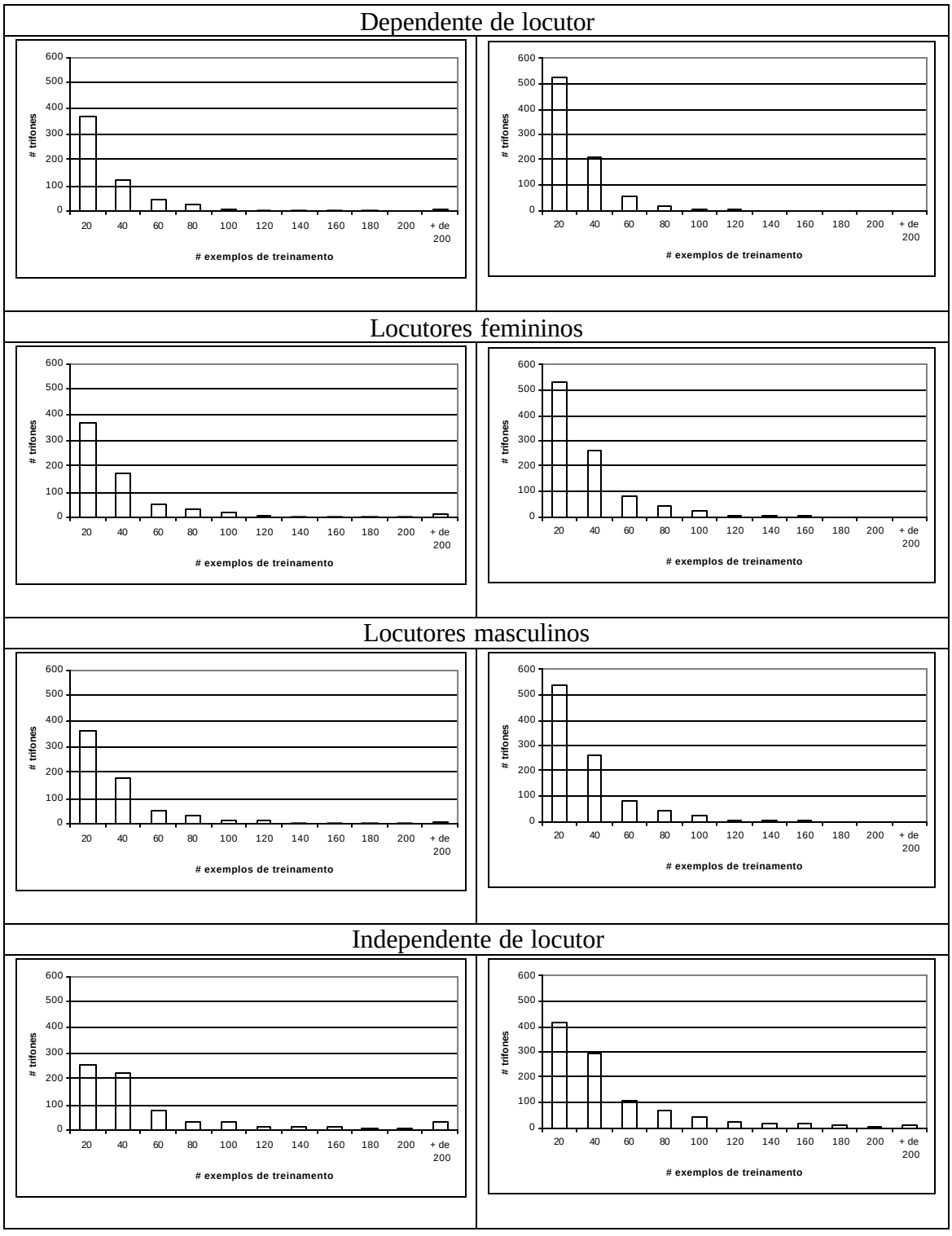


Figura 22: número de exemplos de treinamento para os trifones. Os gráficos da coluna da esquerda referem-se aos trifones gerados através das classes fonéticas, e os da direita, aos trifones gerados a partir da configuração do trato vocal.

7.10.3. Influência dos procedimentos de diminuição dos cálculos necessários na etapa de busca no tempo de reconhecimento

Em relação ao algoritmo *Level Building* foram propostos dois métodos para evitar a parada do processamento quando se atingem máximos locais: um baseado na derivada da curva de verossimilhança, e outro no número de níveis consecutivos em que se verifica queda no valor da verossimilhança.

O primeiro procedimento requer que a queda no valor da verossimilhança seja maior que um determinado limiar para sinalizar a parada do algoritmo. Isto pode fazer com que o processamento continue indefinidamente, se o comportamento de queda da verossimilhança for muito suave. De fato, nos testes realizados, notou-se que para um limiar pequeno (-0,003), o sistema economiza tempo, mas também incorre em muitos erros de deleção e, aumentando-se este limiar (-0,004), estes erros de deleção desaparecem mas, em compensação, o tempo de reconhecimento é quase o mesmo daquele obtido quando se usa um número fixo de níveis, o que indica que não houve quase nenhuma redução no número de cálculos.

O segundo procedimento realiza a parada do processamento utilizando uma informação que pode ser vista como sendo a tendência de queda do valor da verossimilhança: se este valor cai por l níveis consecutivos, é bem provável que o máximo global já tenha sido atingido, e o processo de busca pode ser encerrado. Este parece ser um critério mais robusto para a detecção automática do número de níveis, e os resultados experimentais apresentados na seção 7.6.1 comprovam este fato. Em relação a estes resultados foi observado um fato curioso para os testes realizados com $l = 2$: o número de erros diminuiu em relação aos testes de referência com 15 níveis. Pode-se atribuir esta diminuição a um mero acaso: em algumas frases nas quais o sistema cometeu erros de inserção, o procedimento de parada automática pode ter interrompido a busca em um nível anterior ao máximo global, resultando assim em correções destes erros. Desta forma, não se pode afirmar que este procedimento, além de diminuir o tempo de processamento, tem o poder de diminuir a taxa de erro de palavras.

Em termos de economia de cálculos, o segundo procedimento foi o que conseguiu uma maior redução no tempo de processamento: 23,7% contra 2,5% do primeiro, para um desempenho igual ao do *Level Building* com número fixo de níveis.

Para o algoritmo *One Step* foi testado o procedimento *Beam Search* e conseguiu-se obter uma redução substancial no tempo de processamento sem prejudicar a taxa de acerto de palavras através da escolha de um limiar de poda conveniente. De fato, observou-se uma redução de 46,6% no tempo de processamento, sem deteriorar a taxa de acertos com um limiar $\Delta = 30$. Se for permitida uma pequena queda de desempenho (0,31%), é possível obter uma redução de 51,5% no tempo de processamento, fazendo $\Delta = 25$, o que parece ser uma escolha razoável. Não se conseguiu uma redução no tempo de processamento de uma ordem de grandeza, como reportado na literatura [13], mas espera-se que com uma revisão na implementação do programa este valor venha a ser eventualmente atingido.

Um comentário deve ser feito acerca das implementações dos dois algoritmos de busca: sem utilizar nenhuma otimização, e para um mesmo número de níveis, o *Level Building* apresenta tempos de processamento menores do que o *One Step* (dados apresentados na Tabela 14 para o *Level Building* e Tabela 15 para o *One Step*). Isto se deve à forma de implementação dos códigos, mas não invalida os resultados e as análises.

7.10.4. Influência da transcrição fonética das frases de treinamento no desempenho do sistema.

Os resultados dos testes mostraram que a transcrição padronizada para todas as locuções não afeta de forma significativa o desempenho do sistema. Isto pode ser uma informação valiosa quando se deseja construir grandes bases de dados envolvendo centenas ou milhares de locutores. Entretanto, este resultado precisa ser visto com

cuidado, visto que, novamente, os testes foram realizados com um modelo de linguagem bastante restritivo, o que poderia mascarar o efeito nocivo da transcrição padronizada no desempenho do sistema.

7.10.5. Influência do número de versões de cada palavra no arquivo de vocabulário.

A utilização de um arquivo de vocabulário simplificado, com apenas uma versão para cada palavra, fez com que o tempo de processamento caísse mais de 50 % em todos os casos. Ainda, a taxa de acertos subiu cerca de 1 % para todos os testes.

O primeiro resultado era esperado, uma vez que a diminuiçào do número de palavras no vocabulário corresponde a uma diminuição no espaço de busca e, consequentemente, num menor tempo de reconhecimento.

Já o segundo resultado parece ser estranho, uma vez que com menos versões de cada palavra teríamos um casamento pior das diferentes locuções de entrada com os modelos previstos no vocabulário. Entretanto, é bom lembrar que as versões escolhidas para cada palavra foram as que ocorreram com maior frequência nas locuções da base de dados e, desta forma, pode-se considerar que este vocabulário foi otimizado para estes locutores. O que ajudou bastante neste bom desempenho foi a uniformidade das pronúncias dos locutores de teste, já que a maioria nasceu no estado de São Paulo. Provavelmente ao usarmos este mesmo arquivo de vocabulário com locutores de outras regiões o desempenho do sistema irá cair.

Este resultado vem comprovar as afirmações feitas na seção 6.4.1, de que a tarefa de reconhecimento fica mais difícil à medida que o vocabulário aumenta. Existe então um compromisso entre flexibilidade e perplexidade que deve ser tratado de forma adequada. Uma solução possível seria criar vários arquivos de vocabulário, especializados em várias regiões do país.

7.10.6. Desempenho final do sistema.

Comparando os resultados dos primeiros testes, mostrados na seção 7.4, com os resultados dos testes finais, descritos na seção 7.9, temos o comparativo mostrado na Tabela 21.

Pode-se verificar que a taxa de acertos subiu entre 1,52 %, no caso dependente de locutor, e 3,34 %, para os testes realizados com locutores masculinos, e o tempo de processamento caiu quase 76 % para os testes com dependência de locutor.

Tabela 21: Quadro comparativo do desempenho do sistema nos testes inciais e nos testes finais.

Testes iniciais				Testes finais			
Algoritmo de busca							
Level Building, 15 níveis				*One Step*, 15 níveis, *Beam Search*, $\Delta = 30$			
Vocabulário							
Expandido (1633 palavras)				Simplificado (694 palavras)			
Subunidades fonéticas							
Fones independentes de contexto				Trifones baseados na conf. trato vocal			
% erros de palavra							
Dep.	Masc.	Fem.	Indep.	Dep.	Masc.	Fem.	Indep.
10,50	18,95	21,69	21,31	8,98	15,61	19,63	18,76
Tempo médio de reconhecimento (dep. locutor)							
05:58 minutos				01:26 minutos			

8. Conclusões.

Neste trabalho foram estudados alguns apectos da teoria referente ao reconhecimento de fala, com ênfase especial ao problema de reconhecimento de fala contínua com vocabulário extenso e independência do locutor. Todas as etapas da construção de um sistema completo foram percorridas, desde o projeto e confecção de uma base de dados para treinamento e testes, passando por todas as ferramentas necessárias ao tratamento dos dados, até o desenvolvimento final do sistema. Isto proporcionou uma boa compreensão das questões envolvidas em cada uma das etapas do desenvolvimento de tais sistemas, terminando em um *software* bastante amigável que será utilizado e ampliado em pesquisas futuras. A tempo, o sistema implementado já está sendo utilizado por outros pesquisadores em trabalhos de reconhecimento de dígitos conectados e adaptação ao locutor.

Na confecção da base de dados pôde-se perceber que, em fala contínua, mesmo sendo produzida a partir da leitura de um texto, as coarticulações são bastante fortes. Ainda, a variação de pronúncia e de ritmo de uma mesma palavra devido ao sotaque, nível de educação, e outros fatores é bastante grande. Todos estes fatores contribuem para tornar mais difícil o problema de reconhecimento de fala contínua com independência do locutor. Neste sentido, as técnicas de adaptação ao locutor são de grande importância no sentido de minimizar a amplitude destas variações para os sistemas de reconhecimento.

Os locutores da base de dados foram agrupados de diferentes maneiras de modo a realizar os seguintes testes: independente de locutor, somente locutores femininos, somente locutores masculinos e dependente de locutor. Estes testes têm o objetivo de

investigar a influência do conjunto de locutores no desempenho do sistema. Verificou-se que quando o sistema é utilizado no modo independente de locutor, o seu desempenho cai em relação ao modo dependente do locutor, o que é esperado. Para os testes com dependência de sexo, os testes com locutores masculinos apresentaram os resultados esperados, sendo o desempenho do sistema situado em uma posição intermediária entre aquele observado no modo independente do locutor e no dependente de locutor. A surpresa ficou para os testes com locutores femininos, com um desempenho pior do que aquele observado no modo independente do locutor. Esta discrepância nos resultados parece ter sido causada pela presença de um locutor com o qual o desempenho do sistema foi bastante ruim, distorcendo os resultados. Como a base de dados não é muito grande, um desempenho ruim para um dos locutores influi de forma significativa nos resultados finais.

Em relação às sub-unidades acústicas, foram avaliados três conjuntos: fones independentes de contexto, trifones baseados nas classes fonéticas, e trifones baseados na configuração do trato vocal. Na geração dos modelos trifones, os fones independentes de contexto foram utilizados para inicialização. Após o treinamento, os modelos dos trifones foram mesclados com os modelos dos fones independentes de contexto correspondentes utilizando o procedimento *Deleted Interpolation*.

Não foram utilizados os trifones da forma usual pois o número destes seria muito grande e não haveria dados de treinamento suficientes. Agrupando-se os fones em classes, procurou-se diminuir o número de trifones, tentando manter a consistência, que é a característica interessante destas sub-unidades.

Inicialmente foram feitos testes com os fones independentes de contexto para estabelecer um desempenho padrão para o sistema. Os testes com trifones gerados a partir das classes fonéticas mostraram um ligeiro aumento de desempenho para os modos independente de locutor e para os locutores masculinos, mas apresentaram um resultado pior do que os fones independentes de contexto para os locutores femininos e para os testes com dependência de locutor. Já os testes com os trifones gerados a partir da configuração do trato vocal apresentaram uma melhora do desempenho em todos os casos. Desta forma, pode-se concluir que os trifones baseados na configuração do trato

vocal são unidades consistentes, enquanto que a divisão dos fones nas respectivas classes fonéticas parece não ser uma escolha adequada.

O aumento na taxa de acertos com o uso dos trifones foi bastante pequeno, o que, à primeira vista, não justificaria o seu uso, já que as necessidades de armazenamento aumentam consideravelmente: são 1018 trifones contra 36 fones independentes de contexto. Entretanto, uma análise mais profunda revela o seguinte:

- o modelo de linguagem de pares de palavras utilizado é bastante restritivos, o que poderia estar elevando de forma exagerada o desempenho dos fones independentes de contexto, mascarando o resultado;
- o número de exemplos de treinamento para os trifones é muito pequeno, resultando em sub-unidades extremamente mal treinadas. No procedimento *Deleted Interpolation* este fato fica bastante claro pois, na maioria dos casos, os fones independentes de contexto obtiveram um desempenho melhor do que os trifones.

Com estas considerações, talvez o uso de um modelo de linguagem menos restritivo e uma base de dados maior, que proporcione um treinamento adequado aos trifones, possam mudar este quadro.

Foram propostos dois métodos para diminuir o tempo de processamento para o reconhecimento utilizando o algoritmo *Level Building*. A idéia destes métodos é tentar determinar de forma automática o número de níveis de busca necessários para reconhecer cada locução. O primeiro método baseia-se na informação fornecida pela derivada da curva de verossimilhança e, o segundo, no número de níveis consecutivos em que o valor da verossimilhança cai. O primeiro método conseguiu uma redução de 2,5% no tempo de processamento e, o segundo, 23,7%, sem queda no desempenho.

Foi também testada a técnica *Beam Search* para o algoritmo *One Step*, conseguindo-se uma redução de 46,6% no tempo de processamento, sem alterar a taxa de acertos, e de 51,5% com uma deterioração de apenas 0,31% na taxa de acertos de palavra.

Também foi verificada a influência da precisão da transcrição fonética das locuções de treinamento no desempenho do sistema. Estes mostraram uma deterioração

muito pequena quando se adota uma transcrição fonética padrão todas as pessoas, o que parece indicar que este procedimento possa ser adotado sem maiores problemas. Novamente, o uso de um modelo de linguagem bastante restritivo pode estar mascarando estes resultados, e o efeito nocivo deste procedimento simplificado pode ser um pouco maior. De qualquer forma, este procedimento é adotado em muitos sistemas comerciais (IBM por exemplo) uma vez que uma transcrição fonética criteriosa de cada locutor é uma atividade extremamente tediosa e consome um tempo bastante grande. A possibilidade de se conseguir bases de dados maiores para o treinamento do sistema sem a preocupação de uma transcrição personalizada para cada locutor é um fator que compensa em excesso a pequena degradação no desempenho provocada pela transcrição fonética padronizada..

A falta de grandes bases de dados em português para o treinamento e avaliação parece ser o grande entrave para um desenvolvimento mais rápido e consistente das pesquisas em reconhecimento de fala no Brasil. Infelizmente este trabalho não pode ser feito por uma pessoa ou instituição isolada, mas requer um grande esforço conjunto de órgãos governamentais, iniciativa privada e comunidade científica. De fato, nos EUA e na Europa, houve um grande avanço na tecnologia de voz após a criação de grandes bases de dados, o que permitiu comparar os resultados de forma consistente, e determinar quais idéias são realmente boas, evitando duplicação de esforços.

Aplicando a idéia de uma transcrição fonética padronizada ao arquivo de vocabulário, foi possível reduzir o universo de busca de 1633 palavras para apenas 694, resultando em uma diminuição bastante significativa no tempo de processamento. Como as versões escolhidas para representar cada palavra foram selecionadas a partir das realizações mais comuns observadas na base de dados, conseguiu-se até uma melhora no desempenho do sistema, um fato que não era esperado, mas que pode ser explicado pela menor perplexidade imposta ao sistema de reconhecimento, aliada a modelos que correspondem de fato às locuções apresentadas para o reconhecimento. Espera-se entretanto, que se o sistema for treinado com locutores provenientes de outras regiões, e portanto com formas de pronúncia diferentes, o desempenho do sistema venha a cair. A

construçào de arquivos de vocabulário diferentes para cada região do país parece ser uma alternativa viável para resolver este problema.

Utilizando todas as otimizações apresentadas neste trabalho, a configuração ideal para este sistema seria (para o reconhecimento das frases desta base de dados):

- algoritmo de busca: *One Step* com Viterbi *Beam Search* e limiar de poda $\Delta = 30$.
- modelo de duração de palavras.
- modelo de linguagem de pares de palavras.
- parâmetros mel-cepstrais, com respectivos parâmetros delta e delta-delta.
- vocabulário simplificado, com apenas uma versão para cada palavra.
- subunidades fonéticas: trifones baseados na configuração do trato vocal.

Com estas configurações, o sistema atingiu uma taxa de acertos de 81,24 % no modo independente de locutor, com um tempo médio de reconhecimento por volta de 01:30 minutos em uma máquina com processador AMD-K6 350 MHz com 64 MB de memória RAM.

Como sugestões para trabalhos futuros, pode-se citar o estudo e desenvolvimento de um sistema de reconhecimento de fala baseado em Modelos de Markov Contínuos, e o treinamento destes baseado em critérios discriminativos. Também poderiam ser estudados algoritmos de busca mais velozes como o *Stack Decoder* [24] e o algoritmo Herrmann-Ney. Modelos de linguagem mais avançados tais como gramáticas dependentes de contexto, e métodos de adaptação ao locutor também contribuiriam para a melhoria do desempenho final do sistema

9.Bibliografia.

[1] ALCAIM, A., SOLEWICZ, J. A., MORAES, J. A., Freqüência de ocorrência dos fones e lista de frases foneticamente balanceadas no português falado no Rio de Janeiro. *Revista da Sociedade Brasileira de Telecomunicações*. 7(1):23-41. Dezembro, 1992.

[2] ALLEVA, F., HUANG, X., HWANG, M. Y. An improved search algorithm using incremental knowledge for continuous speech recognition. *Proceedings of ICASSP*. Minneapolis, Minnesota, 1993.

[3] AUBERT, X. NEY, H. Large vocabulary continuous speeech recognition using word graphs. *Proceedings of ICASSP*, Detroit, MI, May 1995.

[4] BAHL, L. R. JELINEK, F., MERCER, R. L. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* PAMI-5(2). March 1983.

[5] BAKER, J. K. The Dragon System – An Overview. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASP-23(4):24-29. February, 1975.

[6] BAKER, J. K. Stochastic modeling for automatic speech understanding. *in Speech Recognition*. Reddy, D. R. ed. New York: Academic. 1975. Pp. 521-542.

[7] BD-PUBLICO (Base de Dados em Português eUropeu, vocaBulário Largo, Independente do orador e fala Contínua) http://www.speech.inesc.pt/bib/Trancoso98a/bdpub.html (31/03/99).

[8] BELLMAN, R. *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[9] CALLOU, D. e LEITE, Y. *Iniciação à fonética e à fonologia*. Rio de Janeiro : Jorge Zahar, 1995.

[10] CALLOU, D. e MARQUES, M.H. Os estudos dialetológicos no Brasil e o projeto de estudo da norma linguística culta. *Littera*, n. 8, 1973. Pp. 100-101.

[11] COLE, R. et al. Corpus development activities at the Center for the Spoken Language Understanding. *Proceedings of the ARPA Workshop on Human Language Technology*. April 7-11, 1994

[12] COLE, R. et al. Telephone speech corpus development at CSLU. *Proceedings of ICSLP*. Yokohama, Japan, September, 1994.

[13] COLE, R. A., ed., *Survey of the State of the Art in Human Language Technology*. http://cslu.cse.ogi.edu/publications/index.htm, (26/10/98).

[14] DAVIS, S. & MELMERTSTEIN, P. Comparison of parametric representations for monossylabic word recognition in continously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASP-28(4):357-366. August, 1980.

[15] DELLER Jr., J. R., PROAKIS, J. G., HANSEN, J.H.L. *Discrete time processing of speech signals*. MacMillan Publishing Company. New York. 1993.

[16] EUROM_1 : a multilingual european speech database. http://www.icp.grenet.fr/Relator/multiling/eurom1.html#PortugCorpus. , (31/03/99)

[17] HAYKIN, Simon. *Neural Networks - A Comprehensive Foundation*. MacMillan Publishing Company. New York 1994.

[18] HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America.* 87(4):1738-1752. 1990.

[19] HERMANSKY, H. Exploring temporal domain for robustness in speech recognition. *Proceedings of the 15th International Congress on Acoustics*, Trondheim, Norway, June 26-30, pp 61-64, 1995.

[20] HOPCROFT, J. E., ULLMAN, J. D. Introduction to Automata Theory, Languages and Computation. Reading, Mass.:Addison-Wesley, 1979.

[21] HOSOM, J. P., COLE, R. A., A diphone-based digit recognition system using neural networks. *Proceedings of the ICASSP*, Munich, German, April, 1997.

[22] HU, Z. et al. Speech recognition using sillable-like units. Proceedings of ICLSP. Philadelphia, October, 1996.

[23] HWANG, M. Y., HUANG, X., Shared-distribution hidden Markov models for speech recognition. *IEEE Transactions on Speech and Audio Processing*. 1(4):414-420. October, 1993.

[24] JELINEK, F. A fast sequential decoding algorithm using a stack. *IBM J. Res. Develop.* vol. 13, pp. 675-685. November. 1969.

[25] JELINEK, F. Language modelling for speech recognition. *Proceedings of the ECAI Workshop*. 1996.

[26] JELINEK, F., BAHL, L. R., and MERCER, R. L. Design of a linguistic statistical decoder for the continuous speech. *IEEE Transactions on Information Theory*. 21:250-256. May, 1975.

[27] JUANG, B.-H. and RABINER, L. R. The segmental K-Means algorithm for estimating parameters if Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing*. VOL ASSP - 38(9):1639-1641. September, 1990.

[28] LEE, C. H. and RABINER, L. R. A frame-syncronous network search algorithm for connected word recognition. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(11):1649-1658. November, 1989.

[29] LEE, K. F., HON, H. W., REDDY, R. An overview of the SPHINX speech recognition system. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(1):35-45. April, 1990.

[30] LEE, K. F. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(4):599-609. April, 1990.

[31] LINDE, Y., BUZO, A., GRAY, R. M. An algorithm for vector quantizer design. *IEEE Transactions on Communications*. COM-28(1). January, 1980.

[32] LOWERRE, B. and REDDY, R. The HARPY speech understanding system. in *Trends in Speech Recognition*. LEA, W. ed. Englewood Cliffs, NJ:Prentice-Hall, 1980. pp. 340-346.

[33] MORAIS, E. S. Reconhecimento automático de fala contínua empregando modelos híbridos ANN+HMM. Tese de Mestrado. UNICAMP. Campinas. 1997.

[34] MYERS, C. S. and LEVINSON, S. E. Speaker-independent connected word recognition using a sintax directed dynamic programming procedure. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-30:561-565. August, 1982.

[35] MYERS, C. S. and RABINER, L. R. A level building dynamic time warping algorithm for connected word recognition. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29:284-297. April, 1981.

[36] NEY, H. The use of a one-stage dynamic programming algorithm for connected word recognition. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(2):263-271. April, 1984.

[37] NEY, H. ESSEN, U., KNESER, R. On structuring probabilistic dependences in stochastic language modelling. Computer Speech and Language. v. 8:1-38. 1994.

[38] NEY, H., MERGEL, D.,NOLL, A. and PAESLER, A. Data driven search organisation for continuous speech recognition. *IEEE Transactions on Signal Processing*, 40(2):272-281, February, 1992.

[39] PALLET, D. S. et al. 1997 BROADCAST NEWS BENCHMARK TEST RESULTS: ENGLISH AND NON-ENGLISH. Proceedings of theDARPA Broadcast News Transcription and Understanding Workshop. February 8-11, 1998. Lansdowne, Virginia

[40] RABINER, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257-286. February, 1989.

[41] RABINER, L. *Fundamentals of speech recognition*. Prentice Hall Press. 1993.

[42] RUDNICKY, A. I., HAUPTMANN, A. G., and LEE, K. F. Survey of Current Speech Technology. http://www.lti.cs.cmu.edu/Research/cmt-tech-reports.html, (22/11/98).

[43] SAKOE, H. Two-level DP-matching – A dynamic programming-based pattern matching algorithm for connected word recognition. . *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-27:588-595. December, 1979.

[44] SANTOS, S. C. B., ALCAIM, A. Inventários reduzidos de unidades fonéticas do português brasileiro para o reconhecimento de voz contínua. *Anais do XIV Simpósio Brasileiro de Telecomunicações*. Recife, Agosto, 1997.

[45] SCHALWYK, J. et al. Embedded implementation of a hybrid neural-network telephone speech recognition system. *Presented at IEEE International Conference on Neural Networks and Signal Processing*. Nanjing, China, December 10-13, pp-800-803. 1995.

[46] SCHWARTZ, R. M., et al. Improved hidden Markov modelling phonemes for continuous speech recognition. *Proceedings of ICASSP*. April 1984.

[47] SIEGLER, M. A. and STERN, R. On the effects of speech rate in large vocabulary speech recognition systems. *Proceedings of the ICASSP*. pp. 612-615, Detroit, MI, May, 1995.

[48] SULLIVAN, T. M. & STERN, R. Multi-microphone correlation-based processing of robust speech recognition. *Proceedings of the ICASSP*. Minneapolis, Minnesota. 1993.

[49] TEBELSKIS, J. *Speech recognition using neural networks*. Pittsburg¸ Pensilvania. PhD Thesis. School of Computer Science. Carnegie Mellon University. 1995.

[50] VINTSYUK, T. K. Element-wise recognition of continuous speech composed of words from a specified dictionary. *Kibernetika.* Vol. 7:133-143. March – April. 1971

[51] WILPON, J. Et al. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*. VOL ASSP-38(11):1870-1878. November, 1990.

[52] YAN, Y., FANTY, M., COLE, R. Speech recognition using neural networks with forward-backward probability generated targets. *Proceedings of the ICASSP*, Munich, April 1997.

[53] YNOGUTI, C. A., MORAIS, E. S., VIOLARO, F. A comparison between HMM and hybrid ANN-HMM based systems for continuous speech recognition. *Proceedings of the International Telecommunications Symposium*. São Paulo. August, 9-13, 1998.

[54] YNOGUTI, C. A., VIOLARO, F. Uma proposta para operação em tempo real de sistemas de reconhecimento de palavras isoladas utilizando redes neurais. *Anais do XIV Simpósio Brasileiro de Telecomunicações*. Recife, Agosto, 1997.

[55] ZHAN, P. et al. Speaker normalization and sepaker adaptation – a combination for conversational speech recognition. *Proceedings of EUROSPEECH*, 1997.

Apêndice A.

Listas de frases utilizadas neste trabalho.

LISTA 01

A questão foi retomada no congresso.
Leila tem um lindo jardim.
O analfabetismo é a vergonha do país.
A casa foi vendida sem pressa.
Trabalhando com união rende muito mais.
Recebi nosso amigo para almoçar.
A justiça é a única vencedora.
Isso se resolverá de forma tranquila.
Os pesquisadores acreditam nessa teoria.
Sei que atingiremos o objetivo.

LISTA 02

Nosso telefone quebrou.
Desculpe se magoei o velho.
Queremos discutir o orçamento.
Ela tem muita fome.
Uma índia andava na mata.
Zé, vá mais rápido!
Hoje dormirei bem.
João deu pouco dinheiro.
Ainda são seis horas.
Ela saía discretamente.

LISTA 03

Eu vi logo a Iôiô e o Léo.
Um homem não caminha sem um fim.
Vi Zé fazer essas viagens seis vezes.
O atabaque do Tito é coberto com pele de gato.
Ele lê no leito de palha.
Paira um ar de arara rara no Rio Real.
Foi muito difícil entender a canção.
Depois do almoço te encontro.
Esses são nossos times.
Procurei Maria na copa.

LISTA 04

A pesca é proibida nesse lago.
Espero te achar bem quando voltar.
Temos muito orgulho da nossa gente.
O inspetor fez a vistoria completa.
Ainda não se sabe o dia da maratona.
Será muito difícil conseguir que eu venha.
A paixão dele é a natureza.
Você quer me dizer a data?
Desculpe, mas me atrasei no casamento.
Faz um desvio em direção ao mar!

LISTA 05

A velha leoa ainda aceita combater.
É hora do homem se humanizar mais.
Ela ficou na fazenda por uma hora.
Seu crime foi totalmente encoberto.
A escuridão da garagem assustou a criança
Ontem não pude fazer minha ginástica.
Comer quindim é sempre uma boa pedida.
Hoje eu irei precisar de você.
Sem ele o tempo flui num ritmo suave.
A sujeira lançada no rio contamina os peixes.

LISTA 06

O jogo será transmitido bem tarde.
É possível que ele já esteja fora de perigo.
A explicação pode ser encontrada na tese.
Meu vôo tinha sido marcado para as cinco.
Daqui a pouco a gente irá pousar.
Estou certo que mereço a atenção dela.
Era um belo enfeite todo de palha.
O comércio daqui tem funcionado bem.
É a minha chance de esclarecer a notícia.
A visita transformou-se numa reunião íntima.

LISTA 07

O cenário da história é um subúrbio do Rio.
Eu tenho ótima razão para festejar.
A pequena nave medirá o campo magnético.
O prêmio será entregue sem sessão solene.
A ação se passa numa cidade calma.
Ela e o namorado vão a Portugal de navio.
O adiamento surpreendeu a mim e a todos
A gente sempre colhe o que plantou.
Aqui é onde existem as flores mais interessantes.
A corrida de inverno aconteceu com vibração.

LISTA 08

Esse empreendimento será de enorme sucesso.
As feiras livres não funcionam amanhã.
Fumar é muito prejudicial à saúde.
Entre com seu código e o número da conta.
Reflita antes e discuta depois.
As aulas dele são bastante agradáveis.
Usar aditivos pode ser desastroso.
O clima não é mau em Calcutá.
A locomotiva vem sem muita carga.
Ainda é uma boa temporada para o cinema.

LISTA 09

Os maiores picos da Terra ficam debaixo d'água.
A inauguração da vila é quarta-feira.
Só vota quem tiver o título de eleitor.
É fundamental buscar a razão da existência.
A temperatura só é boa mais cedo.
Em muitas regiões a população está diminuindo.
Nunca se pode ficar em cima do muro.
Pra quem vê de fora o panorama é desolador.
É bom te ver colhendo flores.
Eu me banho no lago ao amanhecer.

LISTA 10

É fundamental chegar a uma solução comum.
Há previsão de muito nevoeiro no Rio.
Muitos móveis virão às cinco da tarde.
A casa pode desabar em algumas horas.
O candidato falou como se estivesse eleito.
A idéia é falha, mas interessa.
O dia está bom para passear no quintal.
Minhas correspondências não estão em casa.
A saída para a crise dele é o diálogo.
Finalmente o mau tempo deixou o continente.

LISTA 11

Um casal de gatos come no telhado.
A cantora foi apresentar seu último sucesso.
Lá é um lugar ótimo para tomar uns chopinhos.
O musical consumiu sete meses de ensaio.
Nosso baile inicia após as nove.
Apesar desses resultados, tomarei uma decisão.
A verdade não poupa nem as celebridades.
As queimadas devem diminuir este ano.
O vão entre o trem e a plataforma é muito grande.
Infelizmente não compareci ao encontro.

LISTA 12

As crianças conheceram o filhote de ema.
A bolsa de valores ficou em baixa.
O congresso volta atrás em sua palavra.
A médica receitou que eles mudassem de clima.
Não é permitido fumar no interior do ônibus.
A apresentação foi cancelada por causa do som.
Uma garota foi presa ontem à noite.
O prato do dia é couve com atum.
Eu viajarei ao Canadá amanhã.
A balsa é o meio de transporte daqui.

LISTA 13

O grêmio ganhou a quadra de esportes.
Hoje irei à vila sem meu filho.
Essa magia não acontece todo dia.
Será bom que você estude esse assunto.
O menu incluía pratos bem saborosos.
Podia dizer as horas, por favor?
A casa é ornamentada com flores do campo.
A Terra é farta, mas não infinita.
O sinal emitido é captado por receptores.
A mensalidade aumentou mais que a inflação.

LISTA 14

O tele-jornal termina às sete da noite.
A cabine telefônica fica na próxima rua.
Defender a ecologia é manter a vida.
Nesse verão o calor está insuportável.
Um jardim exige muito trabalho.
O mamão que eu comprei estava ótimo
Meu primo falará com a gerência amanhã
De dia apague a luz sempre.
A sociedade uruguaia tem que se mobilizar.
Suas atitudes são bem calmas.

LISTA 15

Dezenas de cabos eleitorais buscavam apoio.
A vitória foi paga com muito sangue.
Nossa filha tem amor por animais.
Esse peixe é mais fatal que certas cobras.
O time continua lutando pelo sucesso.
Essa medida foi devidamente alterada.
O estilete é uma arma perigosa.
Aguarde, quinta eu venho jantar em casa.
A mudança é lenta, porém duradoura.
O clima não é mais seco no interior.

LISTA 16

A sensibilidade indicará a escolha.
A Amazônia é a reserva ecológica do globo.
O ministério mudou demais com a eleição.
Novos rumos se abrem para a informática.
O capital de uma empresa depende da produção.
Se não fosse ela, tudo seria contido.
A principal personagem no filme é uma gueixa.
Receba seu jornal em sua casa.
A juventude tinha que revolucionar a escola.
A atriz terá quatro meses para ensaiar seu canto.

LISTA 17

Muito prazer em conhecê-lo.
Eles estavam sem um bom equipamento.
O sol ilumina a fachada de tarde.
A correção do exame está coerente.
As portas são antigas.
Sobrevoamos Natal acima das nuvens.
Trabalhei mais do que podia.
Hoje eu acordei muito calmo.
Esse canal é pouco informativo.
Parece que nascemos ontem.

LISTA 18

Receba meus parabéns pela apresentação.
Eu planejo uma viagem no feriado.
No lado de cá do rio há uma boa sombra.
A maioria dos visitantes gosta deste monumento.
Minha filha é especialista em música sacra.
A casa só tem um quarto.
A duração do simpósio é de cinco dias.
Ao contrário de nossa expectativa, correu tranquilo.
A intenção é obter apoio do governante.
A fila aumentou ao longo do dia.

LISTA 19

À noite a temperatura deve ir a zero.
A proposta foi inspecionada pela gerência.
O quadro mostra uma face do cotidiano.
Já era bem tarde quando ele me abordou.
O canário canta ao amanhecer.
A lojinha fica bem na esquina de casa.
Meu time se consagrou como o melhor.
Um instituto deve servir a sua meta.
Ele entende quando se fala pausadamente.
Seu saldo bancário está baixo.

LISTA 20

O termômetro marcava um grau.
O discurso de abertura é bem longo.
Eu precisei de microfone na conferência.
Joyce esticou sua temporada até quinta.
Nada como um almoço ao ar livre.
Nossa filha é a primeira aluna da classe.
Gostaria de deitar um pouco.
Não fizemos uma viagem muito cansativa.
Ainda tenho cinco telefonemas para dar.
O hotéis do sudoeste são fantásticos.

Apêndice B.

Resumo informativo dos locutores da base de dados.

Os locutores em destaque foram utilizados para os testes. Os demais, para o treinamento.

LOCUTORES MASCULINOS

locutor	listas	faixa etária	profissão	escolaridade	Cidade/Estado
m01	1 a 4	18 a 60 anos	estudante	superior	São Paulo - SP
m02	5 a 8	18 a 60 anos	engenheiro	superior	Piracicaba – SP
m03	13 a 16	18 a 60 anos	engenheiro	superior	Sta. B. D'Oeste - SP
m04	17 a 20	18 a 60 anos	engenheiro	superior	Fortaleza - CE
m05	9 a 12	18 a 60 anos	engenheiro	superior	S. Caetano do Sul - SP
m06	13 a 16	18 a 60 anos	analista sist.	superior	Ribeirão Preto - SP
m07	1 a 4	18 a 60 anos	professor	superior	São Carlos - SP
m09	9 a 12	18 a 60 anos	estudante	superior	Limeira - SP
m11	13 a 16	18 a 60 anos	estudante	superior	Tupã - SP
m12	17 a 20	18 a 60 anos	estudante	superior	São Paulo – SP
m13	17 a 20	18 a 60 anos	estudante	superior	Santos - SP
m14	1 a 4	18 a 60 anos	comerciante	2º grau	Cotia - SP
m15	17 a 20	18 a 60 anos	engenheiro	superior	Goiânia – GO
m16	5 a 8	18 a 60 anos	estudante	superior	Bauru – SP
m17	9 a 12	18 a 60 anos	estudante	superior	Porto Feliz - SP
m18	5 a 8	18 a 60 anos	estudante	superior	Brasília - DF
m20	1 a 4	18 a 60 anos	estudante	superior	São Paulo – SP
m21	9 a 12	18 a 60 anos	estudante	superior	São Paulo - SP
m23	5 a 8	18 a 60 anos	estudante	superior	Piracicaba - SP
m24	13 a 16	18 a 60 anos	estudante	superior	S. Joaquim da Barra - SP

LOCUTORES FEMININOS

locutor	listas	faixa etária	profissão	escolaridade	Cidade/Estado
f01	17 a 20	18 a 60 anos	engenheira	superior	Barbacena - MG
f02	5 a 8	18 a 60 anos	bibliotecária	superior	São Carlos - SP
f03	9 a 12	18 a 60 anos	estudante	superior	S. Seb. Paraíso - MG
f04	13 a 16	18 a 60 anos	aux. serv. gerais	1º grau	Ribeirão Bonito - SP
f05	17 a 20	18 a 60 anos	analista sistemas	superior	São Carlos - SP
f06	1 a 4	18 a 60 anos	pedagoga	superior	Fortaleza - CE
f07	5 a 8	18 a 60 anos	secretária	superior	São Carlos - SP
f08	9 a 12	18 a 60 anos	secretária	superior	São Carlos - SP
f09	1 a 4	18 a 60 anos	comerciante	2º grau	São Carlos - SP
f10	13 a 16	18 a 60 anos	pedagoga	superior	São Paulo - SP
f11	17 a 20	18 a 60 anos	pedagoga	superior	Vitória - ES
f12	1 a 4	18 a 60 anos	fisioterapeuta	superior	Pindamonhangaba - SP
f13	1 a 4	18 a 60 anos	bióloga	superior	São Paulo - SP
f15	5 a 8	18 a 60 anos	balconista	2º grau	São Carlos - SP
f17	9 a 12	18 a 60 anos	música	superior	Santo André - SP
f18	9 a 12	18 a 60 anos	pedagoga	superior	Camocim S. Félix – PE
f19	13 a 16	18 a 60 anos	estudante	superior	Franca - SP
f20	5 a 8	18 a 60 anos	terap. ocupacional	superior	São Paulo - SP
f21	17 a 20	18 a 60 anos	estudante	superior	Jundiaí - SP
f22	13 a 16	+ 60 anos	aposentada	1º grau	Colatina - ES

Apêndice C.

Dicionário de pronúncias e dados do modelo de duração.

Neste apêndice são mostradas as transcrições fonéticas utilizadas no arquivo de vocabulário simplificado (apenas uma versão para cada palavra).
A estrutura deste dicionário é a seguinte:

transc. gráfica / transc. fonética / média da duração (ms) / desv. padrão da dur.

Nos casos com variância nula (casos em que o modelo foi levantado a partir de uma única amostra), o sistema adota o valor do desvio padrão como 1/3 do valor da média.

, / # / 0 / 0
a / a / 100.771084 / 705.260851
abertura / a b e R t u r a / 490 / 0
abordou / a b o R d o u / 720 / 0
abrem / a b r en / 340 / 0
ação / a s an un / 580 / 0
aceita / a s e y t a / 550 / 0
achar / a x a R / 546.5 / 240.25
acima / a s i m a / 370 / 0
acontece / a k on t E s i / 750 / 0
aconteceu / a k on t e s e u / 640 / 0
acordei / a k o R d e y / 300 / 0
acreditam / a k r e D i t an un / 682 / 0
adiamento / a D i a m e in t u / 790 / 0
aditivos / a D i T i v u s / 620 / 0
agradáveis / a g r a d a v e y s / 800 / 0
aguarde / a g u a R D y / 640 / 0
ainda / a in d a / 376.8 / 3920.96
algumas / a u g u m a z / 550 / 0
almoçar / a u m o s a R / 680 / 0
almoço / a u m o s u / 473 / 49
alterada / a u t e r a d a / 830 / 0
aluna / a l u n a / 286 / 0
amanhã / a m an N an / 567.333333 / 160.888889
amanhecer / a m an N e s e R / 883 / 32400
Amazônia / a m a z o n y a / 790 / 0
amigo / a m i g u / 523 / 0
amor / a m o R / 380 / 0
analfabetismo / a n a u f a b e T i z m u / 1214 / 0
andava / an d a v a / 570 / 0
animais / an n i m a y s / 880 / 0
ano / an n u / 470 / 0
antes / an T y z / 440 / 0
antigas / an T i g a s / 740 / 0
ao / a u / 141.25 / 620.6875
apague / a p a g y / 360 / 0
apesar / a p e s a R / 420 / 0
apoio / a p o y u / 520 / 400
após / a p O z / 320 / 0
apresentação / a p r e z en t a s an un / 870 / 900
apresentar / a p r e z en t a R / 605 / 0
aqui / a k i / 370 / 0
ar / a R / 195 / 3025
arara / a r a r a / 400 / 0
arma / a R m a / 390 / 0
as / a s / 204.1 / 1540.49
assunto / a s un t u / 790 / 0
assustou / a s u s t o u / 500 / 0
à / a / 100.771084 / 705.260851
às / a s / 285 / 0
atabaque / a t a b a k y / 630 / 0
atenção / a t en s an un / 542 / 0
até / a t E / 230 / 0
atingiremos / a T in j i r e m u z / 1000 / 0

atitudes / a T i t u D y s / 590 / 0
atrasei / a t r a s e y / 630 / 0
atrás / a t r a z / 430 / 0
atriz / a t r i s / 410 / 0
atum / a t un / 535 / 0
aulas / a u l a z / 340 / 0
aumentou / a u m en t o u / 720 / 22500
baile / b a y l y / 350 / 0
baixa / b a y x a / 630 / 0
baixo / b a y x u / 460 / 0
balsa / b a u s a / 530 / 0
bancário / b an k a r y u / 395 / 0
banho / b an N u / 625 / 0
bastante / b a s t an T y / 465 / 0
belo / b E l u / 414 / 0
bem / b e in / 317.222222 / 16074.1728
boa / b o a / 305.75 / 3219.1875
bolsa / b o u s a / 446 / 0
bom / b on / 319 / 15289
buscar / b u s k a R / 524 / 0
buscavam / b u s k a v an un / 690 / 0
cabine / k a b i n y / 426 / 0
cabos / k a b u z / 516 / 0
Calcutá / k a u k u t a / 790 / 0
calma / k a u m a / 524 / 0
calmas / k a u m a s / 614 / 0
calmo / k a u m u / 430 / 0
calor / k a l o R / 321 / 0
caminha / k a m in N a / 470 / 0
campo / k an p u / 420 / 12100
Canadá / k a n a d a / 470 / 0
canal / k a n a u / 380 / 0
canário / k a n a r y u / 433 / 0
canção / k an s an un / 654 / 0
cancelada / k an s e l a d a / 700 / 0
candidato / k an D i d a t u / 960 / 0
cansativa / k an s a T i v a / 660 / 0
canta / k an t a / 390 / 0
canto / k an t u / 450 / 0
cantora / k an t o r a / 460 / 0
capital / k a p i t a u / 490 / 0
captado / k a p i t a d u / 816 / 0
carga / k a R g a / 553 / 0
casa / k a z a / 536 / 9061.71429
casal / k a z a u / 403 / 0
casamento / k a z a m en t u / 840 / 0
causa / k a u z a / 360 / 0
cá / k a / 248 / 0
cedo / s e d u / 676 / 0
celebridades / s e l e b r i d a D y s / 890 / 0
cenário / s e n a r y u / 500 / 0
certas / s E R t a s / 440 / 0
certo / s E R t u / 460 / 0
chance / x an s y / 480 / 0
chegar / x e g a R / 490 / 0
chopinhos / x o p in N u s / 870 / 0
cidade / s i d a D y / 440 / 0
cima / s i m a / 347 / 0
cinco / s in k u / 423 / 9317
cinema / s i n e m a / 552 / 0
classe / k l a s y / 486 / 0
clima / k l i m a / 445.333333 / 5763.55556
coberto / k o b E R t u / 490 / 0
cobras / k O b r a s / 720 / 0
código / k O D i g u / 590 / 0
coerente / k o e r en T y / 780 / 0
colhe / k O L y / 330 / 0
colhendo / k o L en d u / 773 / 0
com / k on / 192.666667 / 1424.66667
combater / k on b a t e R / 830 / 0
come / k O m i / 270 / 0
comer / k o m e R / 374 / 0
comércio / k o m E R s y u / 670 / 0
como / k o m u / 250 / 1666.66667
compareci / k on p a r e s i / 620 / 0
completa / k on p l E t a / 840 / 0
comprei / k on p r e y / 360 / 0
comum / k o m un / 480 / 0
conferência / k on f e r en s y a / 734 / 0
congresso / k on g r E s u / 808.5 / 10712.25
conhecê / k on N e s e / 485 / 0
conheceram / k on N e s e r an un / 674 / 0
consagrou / k on s a g r o u / 540 / 0
conseguir / k on s e g i R / 895 / 0
consumiu / k on s u m y u / 476 / 0
conta / k on t a / 510 / 0
contamina / k on t a m i n a / 654 / 0
contido / k on T i d u / 605 / 0
continente / k on T i n en T y / 910 / 0
continua / k on T i n u a / 535 / 0
contrário / k on t r a r y u / 566 / 0
copa / k O p a / 422 / 0
correção / k o rr e s an un / 440 / 0
correspondências / k o rr e s p on d en s y a z / 1154 / 0
correu / k o rr e u / 395 / 0
corrida / k o rr i d a / 395 / 0
cotidiano / k o T i D i an n u / 800 / 0
couve / k o u v y / 455 / 0
criança / k r i an s a / 754 / 0
crianças / k r i an s a s / 704 / 0
crime / k r i m y / 383 / 0
crise / k r i s y / 527 / 0
d'água / d a g u a / 668 / 0
da / d a / 156.666667 / 1493.72222
daqui / d a k i / 402 / 1608
dar / d a R / 270 / 0
das / d a s / 200 / 0

data / d a t a / 600 / 0
de / D y / 128.46875 / 794.561523
debaixo / D y b a y x u / 570 / 0
decisão / d e s i z an un / 740 / 0
defender / d e f en d e R / 480 / 0
deitar / d e y t a R / 400 / 0
deixou / d e y x o u / 530 / 0
dela / d E l a / 507 / 0
dele / d e l y / 455 / 33950
demais / D y m a y s / 480 / 0
depende / d e p en D y / 410 / 0
depois / d e p o y z / 593 / 11449
desabar / d e z a b a R / 600 / 0
desastroso / d e z a s t r o z u / 912 / 0
desculpe / D y s k u p y / 820 / 10000
desolador / d e z o l a d o R / 1048 / 0
desses / d e s y s / 260 / 0
deste / d e s T y / 280 / 0
desvio / d e z v i u / 690 / 0
deu / d e u / 236 / 0
deve / d E v y / 265 / 225
devem / d E v e in / 280 / 0
devidamente / d e v i d a m en T y / 820 / 0
dezenas / d e z e n a z / 500 / 0
dia / D i a / 386 / 7946.66667
diálogo / D i a l o g u / 720 / 0
dias / d i a s / 648 / 0
difícil / D i f i s y u / 720 / 32400
diminuindo / D i m i n u in d u / 998 / 0
diminuir / D i m i n u i R / 480 / 0
dinheiro / D in N e y r u / 640 / 0
direção / D i r e s an un / 700 / 0
discretamente / D i s k r E t a m en T y / 1200 / 0
discurso / D i s k u R s u / 504 / 0
discuta / D i s k u t a / 420 / 0
discutir / D i s k u T i R / 544 / 0
dizer / D i z e R / 410 / 2500
do / d u / 127.421053 / 558.454294
dormirei / d o R m i r e y / 545 / 0
dos / d u s / 230 / 0
duração / d u r a s an un / 494 / 0
duradoura / d u r a d o u r a / 760 / 0
e / y / 81.8333333 / 173.472222
ecologia / e k o l o j i a / 700 / 0
ecológica / e k o l O j i k a / 530 / 0
ela / E l a / 307.4 / 4821.04
ele / e l y / 214 / 1528.4
eleição / e l e y s an un / 470 / 0
eleito / e l e y t u / 580 / 0
eleitor / e l e y t o R / 803 / 0
eleitorais / e l e y t o r a y z / 700 / 0
eles / e l y z / 249 / 81
em / e in / 133.916667 / 1142.07639
ema / e m a / 420 / 0
emitido / e m i T i d u / 660 / 0
empreendimento / in p r e en D i m en t u / 810 / 0
empresa / in p r e z a / 480 / 0
encoberto / in k o b E R t u / 845 / 0
encontrada / in k on t r a d a / 700 / 0
encontro / in k on t r u / 610 / 100
enfeite / in f e y T y / 700 / 0
enorme / e n O R m y / 345 / 0
ensaiar / en s a y a R / 434 / 0
ensaio / in s a y u / 814 / 0
entende / in t en D y / 460 / 0
entender / in t en d e R / 420 / 0
entre / en t r y / 295 / 4225
entregue / in t r E g y / 670 / 0
equipamento / e k i p a m en t u / 800 / 0
era / E r a / 200 / 1600
esclarecer / y s k l a r e s e R / 660 / 0
escola / y s k O l a / 430 / 0
escolha / y s k o L a / 490 / 0
escuridão / y s k u r i d an un / 610 / 0
especialista / y s p e s y a l i s t a / 820 / 0
espero / y s p E r u / 652 / 0
esportes / y s p O R T y s / 1020 / 0
esquina / y s k i n a / 400 / 0
essa / E s a / 300 / 900
essas / E s a s / 350 / 0
esse / e s y / 312 / 1766
esses / e s y s / 410 / 0
estava / y s t a v a / 480 / 0
estavam / y s t a v an un / 600 / 0
está / y s t a / 305.2 / 4836.16
estão / y s t an un / 330 / 0
este / e s T y / 270 / 0
esteja / y s t e j a / 580 / 0
esticou / y s T i k o u / 380 / 0
estilete / y s T i l e T y / 810 / 0
estivesse / y s T i v E s y / 600 / 0
estou / y s t o u / 360 / 0
estude / y s t u D y / 908 / 0
eu / e u / 143.727273 / 3267.28926
exame / y z an m y / 360 / 0
exige / e z i j y / 490 / 0
existem / e z i s t e in / 450 / 0
existência / e z i s t e in s y a / 1140 / 0
expectativa / y s p e k y t a T i v a / 864 / 0
explicação / y s p l i k a s an un / 800 / 0
é / E / 106.697674 / 1240.21092
face / f a s y / 335 / 0
fachada / f a x a d a / 495 / 0
fala / f a l a / 370 / 0
falará / f a l a r a / 530 / 0
falha / f a L a / 575 / 0
falou / f a l o u / 470 / 0

fantásticos / f an t a s T i k u s / 840 / 0
farta / f a R t a / 960 / 0
fatal / f a t a u / 410 / 0
favor / f a v o R / 790 / 0
faz / f a z / 407 / 0
fazenda / f a z en d a / 638 / 0
fazer / f a z e R / 439 / 961
feira / f e y r a / 747 / 0
feiras / f e y r a z / 420 / 0
feriado / f e r i a d u / 800 / 0
festejar / f e s t e j a R / 680 / 0
fez / f e y z / 400 / 0
fica / f i k a / 315.5 / 650.25
ficam / f i k an un / 450 / 0
ficar / f i k a R / 428 / 0
ficou / f i k o u / 368 / 64
fila / f i l a / 415 / 0
filha / f i L a / 398 / 2114.66667
filho / f i L u / 670 / 0
filhote / f i L O T y / 500 / 0
filme / f y u m y / 510 / 0
fim / f in / 410 / 0
finalmente / f i n a u m en T y / 740 / 0
fizemos / f i z E m u z / 476 / 0
flores / f l o r y z / 677.666667 / 77104.2222
flui / f l u y / 343 / 0
foi / f o y / 212.5 / 1502.65
fome / f O m y / 628 / 0
fora / f O r a / 420 / 2500
forma / f O R m a / 443 / 0
fosse / f o s y / 375 / 0
fumar / f u m a R / 554.5 / 9900.25
funcionado / f un s y o n a d u / 748 / 0
funcionam / f un s y o n an un / 540 / 0
fundamental / f un d a m en t a u / 1068 / 144
ganhou / g an N o u / 440 / 0
garagem / g a r a j e in / 564 / 0
garota / g a r o t a / 505 / 0
gato / g a t u / 510 / 0
gatos / g a t u s / 520 / 0
gente / j en T y / 433.333333 / 1622.22222
gerência / j e r en s y a / 548.5 / 1482.25
ginástica / j i n a s T i k a / 820 / 0
globo / g l o b u / 470 / 0
gosta / g O s t a / 510 / 0
gostaria / g o s t a r i a / 460 / 0
governante / g o v e R n an T y / 835 / 0
grande / g r an D y / 400 / 0
grau / g r a u / 220 / 0
grêmio / g r e m y u / 636 / 0
gueixa / g e y x a / 460 / 0
há / a / 100.771084 / 705.260851
história / i s t O r y a / 530 / 0
hoje / o j y / 383.25 / 30739.1875
homem / O m e in / 400 / 0
hora / O r a / 375.5 / 4160.25
horas / O r a s / 651.666667 / 16705.5556
hotéis / o t E y z / 360 / 0
humanizar / u m an n i z a R / 637 / 0
idéia / i d E y a / 420 / 0
ilumina / i l u m i n a / 410 / 0
inauguração / i n a u g u r a s an un / 1050 / 0
incluía / in k l u i a / 780 / 0
índia / in D y a / 456 / 0
indicará / in D i k a r a / 470 / 0
infelizmente / in f e l i z m en T y / 690 / 0
infinita / in f i n i t a / 770 / 0
inflação / in f l a s an un / 620 / 0
informática / in f o R m a T i k a / 780 / 0
informativo / in f o R m a T i v u / 780 / 0
inicia / i n i s i a / 470 / 0
inspecionada / in s p e s y o n a d a / 716 / 0
inspetor / in s p e t o R / 840 / 0
instituto / in s T i t u t u / 510 / 0
insuportável / in s u p o R t a v e u / 1160 / 0
intenção / in t en s an un / 180 / 0
interessa / in t e r E s a / 750 / 0
interessantes / in t e r e s an T y s / 1024 / 0
interior / in t e r i o R / 540 / 3600
íntima / in T i m a / 633 / 0
inverno / in v E R n u / 515 / 0
Iôiô / y o y o / 480 / 0
ir / i R / 70 / 0
irá / i r a / 220 / 0
irei / i r e y / 467.5 / 41820.25
isso / i s u / 410 / 0
jantar / j an t a R / 360 / 0
jardim / j a R D in / 640 / 0
já / j a / 206.5 / 272.25
João / j u an un / 472 / 0
jogo / j o g u / 390 / 0
jornal / j o R n a u / 450 / 3600
Joyce / j O y s y / 420 / 0
justiça / j u s T i s a / 800 / 0
juventude / j u v en t u D y / 620 / 0
lado / l a d u / 311 / 0
lago / l a g u / 568.5 / 992.25
lançada / l an s a d a / 590 / 0
lá / l a / 140 / 0
Leila / l e y l a / 532 / 0
leito / l e y t u / 384 / 0
lenta / l en t a / 520 / 0
leoa / l e o u a / 522 / 0
Léo / l E u / 520 / 0
lê / l e / 281 / 0
lindo / l in d u / 273 / 0
livre / l i v r y / 380 / 0
livres / l i v r y s / 376 / 0

-lo / l u / 220 / 0
locomotiva / l o k o m o T i v a / 780 / 0
logo / l O g u / 500 / 0
lojinha / l O j in N a / 450 / 0
longo / l on g u / 355 / 5625
lugar / l u g a R / 305 / 0
lutando / l u t an d u / 585 / 0
luz / l u s / 322 / 0
magia / m a j i a / 630 / 0
magnético / m a g i n E T i k u / 810 / 0
magoei / m a g u e y / 588 / 0
maiores / m a y O r y s / 720 / 0
maioria / m a y o r i a / 530 / 0
mais / m a y z / 409.444444 / 18657.358
mamão / m a m an un / 410 / 0
manter / m an t e R / 360 / 0
mar / m a R / 380 / 0
maratona / m a r a t o n a / 900 / 0
marcado / m a R k a d u / 624 / 0
marcava / m a R k a v a / 466 / 0
Maria / m a r i a / 503 / 0
mas / m a z / 396.666667 / 7755.55556
mata / m a t a / 700 / 0
mau / m a u / 339 / 17956
me / m y / 137.5 / 1618.75
médica / m E D i k a / 560 / 0
medida / m e D i d a / 520 / 0
medirá / m e D i r a / 390 / 0
meio / m e y u / 330 / 0
melhor / m e L O R / 475 / 0
mensalidade / m en s a l i d a D y / 880 / 0
menu / m e n u / 650 / 0
mereço / m e r e s u / 480 / 0
meses / m e z y s / 439 / 25
meta / m E t a / 380 / 0
meu / m e u / 235 / 4125
meus / m e u s / 241 / 0
microfone / m i k r o f o n y / 650 / 0
mim / m in / 295 / 0
minha / m in N a / 269.333333 / 1410.88889
minhas / m in N a s / 445 / 0
ministério / m i n i s t E r y u / 546 / 0
mobilizar / m o b i l i z a R / 750 / 0
monumento / m o n u m en t u / 786 / 0
mostra / m O s t r a / 370 / 0
móveis / m O v e y z / 840 / 0
mudança / m u d an s a / 460 / 0
mudassem / m u d a s e in / 630 / 0
mudou / m u d o u / 310 / 0
muita / m u y t a / 364 / 36
muitas / m u y t a s / 630 / 0
muito / m u y t u / 319.333333 / 4111.72222
muitos / m u y t u z / 524 / 0
muro / m u r u / 695 / 0
musical / m u z i k a u / 535 / 0
música / m u z i k a / 400 / 0
na / n a / 127 / 951.714286
nada / n a d a / 340 / 0
namorado / n a m o r a d u / 670 / 0
não / n an un / 241.357143 / 6180.94388
nascemos / n a s e m u z / 500 / 0
natal / n a t a u / 390 / 0
natureza / n a t u r e z a / 974 / 0
nave / n a v y / 500 / 0
navio / n a v i u / 500 / 0
nem / n e in / 140 / 0
nessa / n E s a / 310 / 0
nesse / n e s y / 351.5 / 4692.25
nevoeiro / n e v o e y r u / 610 / 0
no / n u / 133.714286 / 785.204082
noite / n o y T y / 455.333333 / 15424.8889
nossa / n O s a / 336.25 / 2350.1875
nosso / n O s u / 312 / 888
nossos / n O s u s / 470 / 0
notícia / n o T i s y a / 753 / 0
nove / n O v y / 440 / 0
novos / n O v u s / 345 / 0
num / n un / 200 / 0
numa / n u m a / 267 / 1089
número / n u m e r u / 304 / 0
nunca / n un k a / 433 / 0
nuvens / n u v en s / 655 / 0
o / u / 99.1666667 / 783.472222
objetivo / o b y j e T i v u / 886 / 0
obter / o b y t e R / 395 / 0
onde / on D y / 300 / 0
ônibus / o n i b u s / 730 / 0
ontem / on t e in / 369 / 848.666667
orçamento / o R s a m en t u / 930 / 0
orgulho / o R g u L u / 620 / 0
ornamentada / o R n a m en t a d a / 840 / 0
os / u s / 170.5 / 583.25
ótima / O T i m a / 433 / 0
ótimo / O T i m u / 480 / 10000
paga / p a g a / 390 / 0
paira / p a y r a / 330 / 0
país / p a i s / 557 / 0
paixão / p a y x an un / 680 / 0
palavra / p a l a v r a / 640 / 0
palha / p a L a / 542 / 25
panorama / p a n o r an m a / 660 / 0
para / p a r a / 264.9 / 10790.29
parabéns / p a r a b e in s / 608 / 0
parece / p a r E s y / 350 / 0
passa / p a s a / 444 / 0
passear / p a s y a R / 483 / 0
pausadamente / p a u z a d a m en T y / 874 / 0
pedida / p e D i d a / 668 / 0

peixe / p e y x y / 484 / 0
peixes / p e y x y s / 740 / 0
pela / p e l a / 215 / 225
pele / p E l y / 240 / 0
pelo / p e l u / 260 / 0
pequena / p y k e n a / 430 / 0
perigo / p y r i g u / 645 / 0
perigosa / p y r i g O z a / 900 / 0
permitido / p e R m i T i d u / 755 / 0
personagem / p e R s o n a j e in / 620 / 0
pesca / p E s k a / 700 / 0
pesquisadores / p e s k i z a d o r y z / 976 / 0
picos / p i k u z / 452 / 0
planejo / p l a n e j u / 575 / 0
plantou / p l an t o u / 550 / 0
plataforma / p l a t a f O R m a / 690 / 0
pode / p O D y / 347.75 / 7538.1875
podia / p u D i a / 460 / 3600
população / p o p u l a s an un / 800 / 0
por / p u R / 200.2 / 924.16
porém / p o r e in / 385 / 0
portas / p O R t a s / 380 / 0
Portugal / p o R t u g a u / 571 / 0
possível / p o s i v e u / 650 / 0
pouco / p o u k u / 465 / 11475
poupa / p o u p a / 312 / 0
pousar / p o u z a R / 686 / 0
pra / p r a / 223 / 0
prato / p r a t u / 514 / 0
pratos / p r a t u z / 720 / 0
prazer / p r a z e R / 341 / 0
precisar / p r e s i z a R / 580 / 0
precisei / p r e s i z e y / 530 / 0
prejudicial / p r e j u D i s i a u / 690 / 0
prêmio / p r e m y u / 504 / 0
presa / p r e z a / 480 / 0
pressa / p r E s a / 671 / 0
previsão / p r e v i z an un / 860 / 0
primeira / p r i m e y r a / 410 / 0
primo / p r i m u / 390 / 0
principal / p r in s i p a u / 504 / 0
procurei / p r o k u r e y / 506 / 0
produção / p r o d u s an un / 552 / 0
proibida / p r o i b i d a / 820 / 0
proposta / p r o p O s t a / 486 / 0
próxima / p r O s i m a / 570 / 0
pude / p u D y / 276 / 0
quadra / k u a d r a / 605 / 0
quadro / k u a d r u / 343 / 0
quando / k u an d u / 316.666667 / 3488.88889
quarta / k u a R t a / 515 / 0
quarto / k u a R t u / 636 / 0
quatro / k u a t r u / 360 / 0
que / k y / 129.928571 / 1970.63776
quebrou / k e b r o u / 570 / 0
queimadas / k e y m a d a z / 546 / 0
quem / k e in / 315 / 4761
quer / k E R / 280 / 0
queremos / k e r e m u z / 456 / 0
questão / k e s t an un / 612 / 0
quindim / k in D in / 592 / 0
quinta / k in t a / 400 / 1600
quintal / k in t a u / 570 / 0
rápido / rr a p i d u / 620 / 0
rara / rr a r a / 454 / 0
razão / rr a z an un / 518 / 169
real / rr e a u / 540 / 0
receba / rr e s e b a / 514.5 / 0.25
recebi / rr e s e b y / 685 / 0
receitou / rr e s e y t o u / 660 / 0
receptores / rr e s e p y t o r y s / 1075 / 0
reflita / rr e f l i t a / 620 / 0
regiões / rr e j i on y z / 684 / 0
rende / rr en D y / 404 / 0
reserva / rr e z E R v a / 440 / 0
resolverá / rr e z o u v e r a / 790 / 0
resultados / rr e z u t a d u s / 750 / 0
retomada / rr e t o m a d a / 726 / 0
reunião / rr e u n i an un / 560 / 0
revolucionar / rr e v o l u s y o n a R / 690 / 0
rio / rr y u / 332.4 / 2639.04
ritmo / rr i T y m u / 470 / 0
rua / rr u a / 436 / 0
rumos / rr u m u s / 450 / 0
sabe / s a b y / 665 / 0
saborosos / s a b o r O z u s / 1230 / 0
sacra / s a k r a / 585 / 0
saía / s a i a / 680 / 0
saída / s a i d a / 515 / 0
saldo / s a u d u / 320 / 0
sangue / s an g y / 590 / 0
são / s an un / 196 / 693.333333
saúde / s a u D y / 604 / 0
se / s y / 174.230769 / 1153.86982
seco / s e k u / 323 / 0
sei / s e y / 416 / 0
seis / s e y z / 341 / 441
sem / s e in / 284.285714 / 5699.63265
sempre / s en p r y / 448 / 8754.66667
sensibilidade / s en s i b i l i d a D y / 910 / 0
ser / s e R / 251 / 1521
será / s e r a / 392.4 / 14071.04
seria / s e r i a / 290 / 0
servir / s e R v i R / 450 / 0
sessão / s e s an un / 440 / 0
sete / s E T y / 382.5 / 6.25
seu / s e u / 248.5 / 4244.58333
sido / s i d u / 330 / 0

simpósio / s in p O z y u / 739 / 0
sinal / s i n a u / 420 / 0
sobrevoamos / s o b r e v o an m u z / 757 / 0
sociedade / s o s i e d a D y / 720 / 0
sol / s O u / 275 / 0
solene / s o l e n y / 570 / 0
solução / s o l u s an un / 590 / 0
som / s on / 480 / 0
sombra / s on b r a / 590 / 0
só / s O / 305 / 7338.66667
sua / s u a / 235 / 1625
suas / s u a z / 350 / 0
suave / s u a v y / 803 / 0
subúrbio / s u b u R b y u / 580 / 0
sucesso / s u s E s u / 710 / 1266.66667
sudoeste / s u d o E s T y / 610 / 0
sujeira / s u j e y r a / 512 / 0
surpreendeu / s u R p r e en d e u / 610 / 0
tarde / t a R D y / 528.75 / 5779.6875
te / T y / 188.666667 / 4803.55556
telefone / t e l e f o n y / 592 / 0
telefonemas / t e l e f o n e m a s / 680 / 0
telefônica / t e l e f o n i k a / 787 / 0
tele- / t E l e / 290 / 0
telhado / t e L a d u / 540 / 0
tem / t e in / 220.333333 / 587.222222
temos / t e m u z / 424 / 0
temperatura / t en p e r a t u r a / 859 / 41616
tempo / t en p u / 375 / 225
temporada / t en p o r a d a / 563 / 1849
tenho / t e N u / 285 / 4225
teoria / t e o r i a / 660 / 0
terá / t e r a / 250 / 0
termina / t e R m i n a / 430 / 0
termômetro / t e R m o m e t r u / 624 / 0
terra / t E rr a / 379 / 6241
tese / t E z y / 574 / 0
time / T i m y / 320 / 400
times / T i m y s / 750 / 0
tinha / T i N a / 375 / 625
Tito / T i t u / 450 / 0
título / T i t u l u / 545 / 0
tiver / T i v E R / 430 / 0
todo / t o d u / 327.5 / 156.25
todos / t o d u s / 660 / 0
tomar / t o m a R / 330 / 0
tomarei / t o m a r e y / 444 / 0
totalmente / t o t a u m en T y / 730 / 0
trabalhando / t r a b a L an d u / 804 / 0
trabalhei / t r a b a L e y / 410 / 0
trabalho / t r a b a L u / 560 / 0
tranquila / t r an k u y l a / 830 / 0
tranquilo / t r an k u y l u / 750 / 0
transformou / t r an s f o R m o u / 711 / 0
transmitido / t r an z m i T i d u / 727 / 0
transporte / t r an s p o R T y / 600 / 0
trem / t r e in / 214 / 0
tudo / t u d u / 280 / 0
último / u T i m u / 320 / 0
um / un / 134.125 / 1014.23438
uma / u m a / 206.714286 / 2443.20408
união / u n i an un / 503 / 0
única / u n i k a / 367 / 0
uns / un s / 155 / 0
uruguaia / u r u g u a y a / 616 / 0
usar / u z a R / 380 / 0
valores / v a l o r y s / 610 / 0
vá / v a / 405 / 0
vão / v an un / 266.5 / 42.25
velha / v E L a / 360 / 0
velho / v E L u / 542 / 0
vem / v e in / 200 / 0
vencedora / v en s e d o r a / 870 / 0
vendida / v en D i d a / 577 / 0
venha / v e N a / 470 / 0
venho / v e N u / 230 / 0
ver / v e R / 565 / 0
verão / v e r an un / 412 / 0
verdade / v e R d a D y / 530 / 0
vergonha / v e R g on N a / 493 / 0
vezes / v e z y s / 570 / 0
vê / v e / 316 / 0
vi / v i / 195 / 25
viagem / v i a j e in / 512 / 19044
viagens / v i a j e in s / 540 / 0
viajarei / v i a j a r e y / 600 / 0
vibração / v i b r a s an un / 575 / 0
vida / v i d a / 520 / 0
vila / v i l a / 541.5 / 2652.25
virão / v i r an un / 505 / 0
visita / v i z i t a / 500 / 0
visitantes / v i z i t an T y s / 790 / 0
vistoria / v i s t o r i a / 746 / 0
vitória / v i t O r y a / 623 / 15129
você / v o s e / 457.666667 / 5830.88889
volta / v O u t a / 400 / 0
voltar / v o u t a R / 600 / 0
vôo / v o u / 516 / 0
vota / v O t a / 516 / 0
zero / z E r u / 436 / 0
zé / z E / 348.5 / 4

Apêndice D.

Algumas frases reconhecidas.

Estas frases foram selecionadas a partir dos resultados dos testes realizados na seção 7.5.2.

lista 1, frase 1	
original	A questão foi retomada no congresso
ind. locutor (f13)	, questão foi retomada no congresso ,
ind locutor (m01).	, cá , questão foi retomada no congresso ,
dep. sexo (f13)	, questão foi retomada no congresso ,
dep. sexo (m01)	, cá , questão foi retomada no congresso ,
dep. locutor	, a questão foi retomada no congresso ,
lista 2, frase 2	
original	Desculpe se magoei o velho.
ind. locutor (f13)	, , desculpe , cinema , meio , e o velho ,
ind locutor (m01).	, desculpe se magoei o , velho ,
dep. sexo (f13)	, , desculpe se magoei o comércio ,
dep. sexo (m01)	, desculpe se magoei o , tese ,
dep. locutor	, desculpe se magoei o velho ,
lista 3, frase 3	
original	Vi Zé fazer essas viagens seis vezes.
ind. locutor (f13)	, a , vi zé , nave , essas viagens , seis vezes , ,
ind locutor (m01).	, vi zé fazer essas , viagens seis vezes
dep. sexo (f13)	, pude , devem , nave , essas viagens , entender , ,
dep. sexo (m01)	, vi zé fazer essas , viagens seis , desses
dep. locutor	, vi zé fazer essas viagens seis vezes ,

lista 4, frase 4

original	O inspetor fez a vistoria completa.
ind. locutor (f13)	, o inspetor , de , e a vistoria completa ,
ind locutor (m01).	, o inspetor fez a vistoria completa ,
dep. sexo (f13)	, inspetor fez a vistoria completa ,
dep. sexo (m01)	, o inspetor fez a vistoria completa ,
dep. locutor	, o inspetor fez a vistoria completa ,

lista 5, frase 5

original	A escuridão da garagem assustou a criança.
ind. locutor (f20)	, cá , escuridão da garagem assustou a criança ,
ind locutor (m23).	, escuridão da garagem assustou a criança ,
dep. sexo (f20)	, cá , a escuridão da garagem assustou a criança ,
dep. sexo (m23)	, escuridão da garagem assustou a criança ,
dep. locutor	, e a escuridão da garagem assustou a criança ,

lista 6, frase 6

original	Estou certo que merço a atenção dela.
ind. locutor (f20)	, isso , certo que mereço a atenção dela ,
ind locutor (m23).	, estou certo que mereço a atenção dela ,
dep. sexo (f20)	, estou certo que mereço a atenção dela ,
dep. sexo (m23)	, estou certo que mereço a atenção dela ,
dep. locutor	, estou certo que mereço a atenção dela ,

lista 7, frase 7

original	O adiamento surpreendeu a mim e a todos.
ind. locutor (f20)	, o adiamento surpreendeu a mim , irá , todos ,
ind locutor (m23).	, orçamento , surpreendeu a mim e a todos
dep. sexo (f20)	, o adiamento surpreendeu a minha , irá , todos ,
dep. sexo (m23)	, o adiamento , surpreendeu a mim , de gatos
dep. locutor	, o adiamento surpreendeu a mim e a todos ,

lista 8, frase 8

original	O clima não é mau em Calcutá.
ind. locutor (f20)	, explicação , é mau em , Calcutá ,
ind locutor (m23).	, sua , clima , não é mau , seis , Calcutá ,
dep. sexo (f20)	, os , clima , verão o mamão , em Calcutá ,
dep. sexo (m23)	, som , clima , não é mau tempo , se , Calcutá ,
dep. locutor	, o clima não é mau em Calcutá ,

lista 9, frase 9

original	É bom te ver colhendo flores.
ind. locutor (f18)	, é bom , estivesse , colhendo flores
ind locutor (m17).	, é bom te ver colhendo flores ,
dep. sexo (f18)	, é bom , tiver , colhendo flores ,
dep. sexo (m17)	, é bom te ver colhendo flores ,
dep. locutor	, até , bom te ver colhendo flores ,

lista 10, frase 10

original	Finalmente o mau tempo deixou o continente
ind. locutor (f18)	, finalmente o mau tempo deixou o continente ,
ind locutor (m17).	, finalmente o mau tempo deixou o continente ,
dep. sexo (f18)	, finalmente o mau tempo deixou o continente ,
dep. sexo (m17)	, finalmente o mau tempo deixou o continente ,
dep. locutor	, finalmente o mau tempo deixou o continente ,

lista 11, frase 1

original	Um casal de gatos come no telhado.
ind. locutor (f18)	, som , casal de dia , dos , come no feriado ,
ind locutor (m17).	, e , casal de gatos come no telhado ,
dep. sexo (f18)	, todos , casal , de dia , dos , come no feriado ,
dep. sexo (m17)	, ficar , saúde , de gatos come no telhado ,
dep. locutor	, som , casal de gatos come no telhado ,

lista 12, frase 2

original	A bolsa de valores ficou em baixa.
ind. locutor (f18)	, a bolsa de valores ficou em baixa ,
ind locutor (m17).	, a bolsa de valores ficou em baixa ,
dep. sexo (f18)	, a bolsa de valores ficou em baixa ,
dep. sexo (m17)	, a bolsa de valores ficou em baixa , se ,
dep. locutor	, a bolsa de valores ficou em baixa ,

lista 13, frase 3

original	Essa magia não acontece todo dia.
ind. locutor (f10)	, essa magia não acontece todo de ,
ind locutor (m11).	, pressa , magia não acontece todo dia ,
dep. sexo (f10)	, essa magia não acontece todo de ,
dep. sexo (m11)	, essa magia não acontece todo dia ,
dep. locutor	, essa magia não acontece todo dia ,

lista 14, frase 4

original	Nesse verão o calor está insuportável.
ind. locutor (f10)	, nesse verão , o calor está insuportável ,
ind locutor (m11).	, nesse verão o calor está insuportável ,
dep. sexo (f10)	, nesse verão , o calor está insuportável ,
dep. sexo (m11)	, nesse verão o calor está insuportável ,
dep. locutor	, , , nesse verão o tempo , calor está insuportável ,

lista 15, frase 5

original	O time continua lutando pelo sucesso.
ind. locutor (f10)	, o time continua lutando pelo sucesso ,
ind locutor (m11).	, por , time continua lutando pelo sucesso ,
dep. sexo (f10)	, o time continua lutando pelo sucesso ,
dep. sexo (m11)	, com , time continua lutando pelo sucesso ,
dep. locutor	, do , time continua lutando pelo sucesso ,

lista 16, frase 6

original	Se não fosse ela, tudo seria contido.
ind. locutor (f10)	, se não fosse ela tudo seria contido ,
ind locutor (m11).	, se não fosse ela tudo seria contido ,
dep. sexo (f10)	, sinal , fosse ela tudo seria contido ,
dep. sexo (m11)	, se não fosse ela tudo seria , cotidiano ,
dep. locutor	, se não fosse ela tem , tudo seria contido ,

lista 17, frase 7

original	Trabalhei mais do que podia.
ind. locutor (f05)	, trabalhei mais que podia ,
ind locutor (m15).	, trabalhei mais que podia ,
dep. sexo (f05)	, trabalhei mais do Tito , tiver ,
dep. sexo (m15)	, trabalhei mais que podia ,
dep. locutor	, trabalhei mais do que podia ,

lista 18, frase 8

original	Ao contrário de nossa expectativa, correu tranquilo.
ind. locutor (f05)	, ao contrário de nossa , expectativa correu tranquilo ,
ind locutor (m15).	, são , contrário de nossa expectativa correu tranquilo ,
dep. sexo (f05)	, ao contrário de nossa , expectativa correu tranquilo ,
dep. sexo (m15)	, ao contrário de nossa expectativa correu tranquilo ,
dep. locutor	, Calcutá , de nossa expectativa , correu tranquilo ,

lista 19, frase 9

original	Ele entende quando se fala pausadamente.
ind. locutor (f05)	, em , tem que , quando se fala pausadamente ,
ind locutor (m15).	, ele entende quando se fala pausadamente ,
dep. sexo (f05)	, crime , quem , quando se fala pausadamente ,
dep. sexo (m15)	, ele entende quando se fala pausadamente ,
dep. locutor	, ele entende quando se fala pausadamente , ,

lista 20, frase 10

original	Os hotéis do sudoeste são fantásticos.
ind. locutor (f05)	, os hotéis do sudoeste são fantásticos ,
ind locutor (m15).	, os hotéis , sudoeste são fantásticos
dep. sexo (f05)	, os hotéis do sudoeste são fantásticos ,
dep. sexo (m15)	, usar , presa , sudoeste são fantásticos
dep. locutor	, dos , hotéis do sudoeste são fantásticos ,