Universidade Estadual do Sudoeste da Bahia - UESB

Departamento de Ciências Exatas - DCE

Curso de Ciência da Computação

Hesdras Oliveira Viana

Construção do modelo de voz para o reconhecimento automático da fala em português brasileiro

Vitória da Conquista, BA

2011

Hesdras Oliveira Viana

Construção do modelo de voz para o reconhecimento automático da fala em português brasileiro

Monografia de conclusão de curso apresentada ao Colegiado de Ciência da Computação - UESB - como parte dos requisitos para obtenção do título Bacharel em Ciência da Computação.

Área de Concentração: Sistemas Inteligentes

Orientador: Prof. Dr. Roque Mendes Prado Trindade

Vitória da Conquista, BA

2011

Monografia de Projeto Final de Graduação com o título *"Construção do modelo de voz para o reconhecimento automático da fala em português brasileiro"*, defendida por Hesdras Oliveira Viana e aprovada em 14 de janeiro de 2011, em Vitória da Conquista, Estado da Bahia, pela banca examinadora constituída pelos professores:

Prof. Dr. Roque Mendes Prado Trindade
Orientador

Profa. Dra. Vera Pacheco
Universidade Estadual do Sudoeste da Bahia

Profa. Dra. Alzira Ferreira da Silva
Núcleo de Tecnologia Educacional

Dedico esse trabalho aos meus pais, Nivaldo e Miralva, aos meus irmãos, Rondinelli e Nivaldo Júnior, a minha tia, Valda, a minha namorada, Karla, a minha prima, Roberta. A todos eu dedico.

Agradecimento

Agradeço primeiramente a Deus, que me deu sabedoria e força para seguir em frente sempre mostrando-me o caminho correto. Obrigado Meu Pai Celestial pela dádiva!

Agradeço aos meus pais, Nivaldo Morais Viana e Miralva Santos de Oliveira Viana, por fazerem sacrifícios inimagináveis em prol da minha educação. Não tenho palavras para agradecê-los meus heróis. Muito obrigado!

Agradeço as minhas tias e tios, em especial a Valda Santana e Raimundo Santana, pelo apoio e vibração inconteste. Tia e tio obrigado por existirem em minha vida!

Agradeço ao meus irmãos, Rondinelli Oliveira Viana e Nivaldo Morais Viana Júnior, pelos conselhos sábios. Obrigado irmaozões!

Agradeço a minha namorada, Karla Conceição Abobreira, pelo carinho e paciência ao longo da minha jornada. Obrigado Amor!

Agradeço ao meus primos e primas, em especial a Roberta Santana, por acreditar no sucesso dessa minha jornada. Obrigado Zelda!

Agradeço a secretária do colégiado de Ciência da Computação, Celina, por me receber sempre com brilho nos olhos, ajudando-me nos momentos chave na minha graduação. Obrigado Cel, nunca te esquecerei!

Agradeço ao meu Orientador, Roque Mendes Prado Trindade, por ajudar-me a concretizar meus sonhos. Obrigado Roque, estamos sempre juntos!

Agradeço ao meus colegas, Jadson, Hilário, Poções (Diogo), Doug, Marcos, Ramom, Elias, Gabriel, Dino, Shintia, Henrique, Esdras, Marlovich, por terem um papel fundamental em minha graduação. Sem vocês não conseguiria terminar essa jornada. Muito Obrigado “Tropa de Elite”!.

Agradeço ao grupo de pesquisa, SIAC, pelos estudos na área de reconhecimento de voz. Obrigado meus mestres!

Agradeço a todos que direto ou indiretamente contribuiu na minha graduação. Muito Obrigado a todos vocês!

“Vive-se de acreditar.
Acredita-se na amizade, na religião, na ciência.
Acredita-se no que não se vê.
Mas, de que valeria o amanhã?
Se não acreditar em você.

O tempo está informatizado
A época sintonizada
Espaço ligado, tocado
Tudo digitado, sincronizado
Coisa imaginária
Fetiche humanizado.

Motivos não faltam
Para viver a alegria
Amar a liberdade
Curtir a natureza
Sentir o calor do sol
O cheiro bom da terra molhada.
Abraçar os amigos
E dizer-lhes: VALEU AMIGO!”

Miralva Santos de Oliveira Viana

Resumo

O reconhecimento automático de voz, tornou-se uma das principais pesquisas na área de computação, criando uma nova interface computador-usuário. Um reconhecedor só obterá sucesso, se a modelagem da voz for feita seguindo os princípios da fonética e fonologia. Para tal, precisa-se definir: gramática, dicionário, unidades a utilizar e treinabilidade.

Este trabalho mostra como construir um modelo de voz em português brasileiro, utilizando ferramentas como o HTK, moldada integralmente em plataforma livre. Também foi feito um programa para reconhecer palavras isoladas, integrando o modelo de voz em português com o engine Julius, treinado sobre as premissas do modelo oculto de Markov, fazendo uso de unidades menores do que a palavra.

Palavras-chave: Reconhecimento de voz, Hidden Markov Model, Modelo Acústico em Português, Julius, Hidden Markov Model Toolkit, Mel-Frequência Cepstrais Coeficiente.

Abstract

The automatic speech recognition, has become a major research area in computing, creating a new computer-user interface. A recognizer will succeed only if the modeling of the voice is done by following the principles of phonetics and phonology. To this end, we need to define: grammar, dictionary, units to be used and trainability.

This paper shows how to construct a model of voice in Brazilian Portuguese, using tools such as HTK, shaped entirely free platform. It was also made a program to recognize individual words, integrating the model voice in Portuguese with the engine Julius, trained on the premises of the hidden Markov model, using units smaller than word.

Key-words: Speech Recognition, Hidden Markov Model, Acoustic Model in Portuguese, Julius, Hidden Markov Model Toolkit, Mel-Frequency Cepstral Coefficients.

Lista de Figuras

Lista de Tabelas

Glossário

GPL - Licença Pública Geral

RAF - Reconhecimento Automático da Fala

HMM - Modelo Oculto de Markov

HTK - Hidden Markov Model Toolkit

MFCC - Mel-Frequência Cepestrais Coeficientes

S.O. - Sistema Operacional

BNF - Forma Normal de Backus

Sumário

1 Introdução

Os avanços tecnológicos vêm convergindo para a substituição do teclado e mouse. O Reconhecimento automático da fala é o principal alvo desses estudos que já inspirou até os produtores de Hollywood em filmes de ficção científica (RABINER; JUANG, 1993).

O reconhecimento da fala é o processo de codificação do sinal da voz que serve como entrada para o computador identificar as palavras ditas e então processá-las (COOK, 2002).

Segundo Bresolin (2008), o processamento da fala é divida em: codificação da fala, síntese da fala e reconhecimento automático da fala. Todas elas convergem para a necessidade de um bom modelo de voz.

A codificação da fala é feita através de técnicas que buscam representar de forma compacta o sinal da voz. Além disso, deve-se perceber não só a integibilidade do que é ouvido, mas também as outras informações como a entonação e a emoção do interlocutor (CARVALHO; DIAS, 2000).

A síntese da fala constitui em produzir sons parecidos com a voz humana a partir de um texto escrito, verificando aspectos como: naturalidade, releva até que ponto o sintetizador soa como a voz humana, e integibilidade, avalia a facilidade do entendimento da saída da fala.

O reconhecimento automático da fala (RAF) refere-se ao aspecto de como a máquina vai reconhecer a fala humana produzindo resultados esperados. O mesmo é dividido em: modo dependente ou independente de locutor. (BRESOLIN, 2008).

No modo dependente de locutor, as elocuções são pronunciadas por locutores previamente conhecidos. Já o modo independente de locutor, qualquer pessoa pode pronunciar frases sem necessidade de treinamento adicional do sistema (JÚNIOR, 2009).

Toda esse processamento da fala necessita de um modelo de voz que seja constituído por: modelo acústico, linguístico e gramatical.

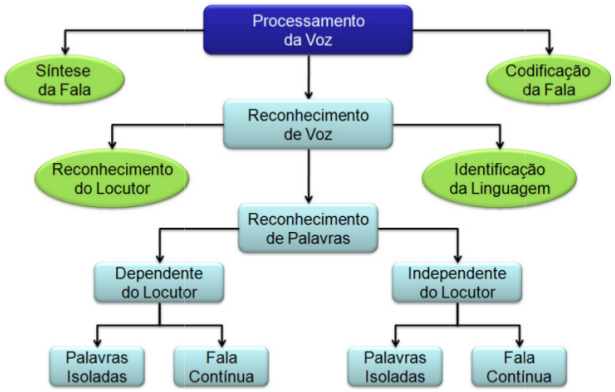

Fig. 1.1: Esquema hierárquico dos sistemas de processamento de voz. (BRESOLIN, 2008).

Esses modelos fazem necessários, pois é a partir deles que é feita a decomposição de sentença em palavras e essas em unidades sub-léxicas como: sílabas, polifones, fones ou unidades sub-fônicas, que é a base de dados para o reconhecimento da fala (MORAIS, 1997).

O uso de ferramentas adaptativas estão em expansão. Hoje, já é possível observar vários programas que seguem a licença pública geral (GPL), que rege os software livres, como por exemplo o programa leitor de tela Orca.

O reconhecimento da fala adentra no processo de inclusão da população com deficiência visual e/ou com problemas de tetraplegia, fornecendo uma nova interface computador-usuário.

Segundo Martins (1997), muitos reconhecedores já foram criados para sanar o problema da acessibilidade, porém, a maioria baseia-se no modelo de voz da língua inglesa, como por exemplo: SPHINX, Byblos e Tangora.

Com base neste contexto, o trabalho em questão tem como objetivo a construção de um modelo de voz em português utilizando ferramentas livres, analisando as técnicas de elaboração de modelos acústico, linguístico e gramatical, integrando com o decodificador Julius.

1.1 Metodologia

O método de abordagem será dedutivo, pois segundo Gil (2002), esse método sai do geral para o particular, fazendo uso de cadeias de raciocínio em ordem descendente.

A tipologia da pesquisa é exploratória, pois segundo o mesmo autor, essa tipologia envolve le-

vantamento bibliográfico e estudos de casos.

A seleção do material da pesquisa foi feita através de livros e na base de dados do Google Acadêmico.

Foram utilizados os seguintes descritores para o levantamento de documentos bibliográficos: reconhecimento; voz; modelo; Hidden Markov Model; fala, no idioma português e inglês. Os documentos selecionados foram: artigos, livros, monografias, dissertações de mestrado e doutorado, sendo que estes estudos se enquadravam entre o ano de 1991 a 2010. Como critério de exclusão encontra-se: documentos que utilizam exclusivamente software proprietário e estudos que não se enquadram no escopo do trabalho.

1.2 Estrutura e Conteúdo do Trabalho

O presente trabalho está estruturado de forma a ilustrar a teoria envolvida no estudo do tema, construção do modelo de voz para o reconhecimento automático da fala em português.

O capítulo 2 aborda o estado da arte, discutindo sobre os métodos utilizados para a construção de um modelo de voz, pautado sobre as premissas da fonética e fonologia.

O capítulo 3 apresenta a forma escolhida neste trabalho para a construção do modelo de voz. Esse capítulo mostra o passo-a-passo da integração do modelo de voz com o decodificador Julius, criando um sistema de reconhecimento de voz por comandos.

O capítulo 4 apresenta a conclusão dos estudos e as sugestões para os trabalhos futuros.

2 *Fundamentação teórica*

Neste capítulo serão abordados conceitos essenciais para a construção de um modelo de voz.

Primeiramente, serão abordados conceitos da formação do som, explicando sua particularidade e representação perante a fonética e fonologia do português brasileiro. Logo após, serão explanadas as diferenças entre os três modelos que norteiam a construção do modelo de voz: acústico, linguístico e gramatical. Por fim, serão mostradas as técnicas e ferramentas para a construção do modelo de voz.

2.1 Produção da voz

A fala constitui o meio mais completo da comunicação entre as pessoas. É através do som que vinculamos significados e interagimos socialmente, sem dar conta de sua organização interna (BISOL, 2005).

Para entendermos como construir um modelo de voz, precisamos nos atentar em como a fala é produzida. Infelizmente não temos um órgão específico para a produção da mesma e sim utilizamos três grupos de órgão que nos apoiam para essa produção, são eles: Sistema Respiratório, Sistema Fonatório e Sistema Articulatório. A figura 2.1 apresenta os três sistemas citados.

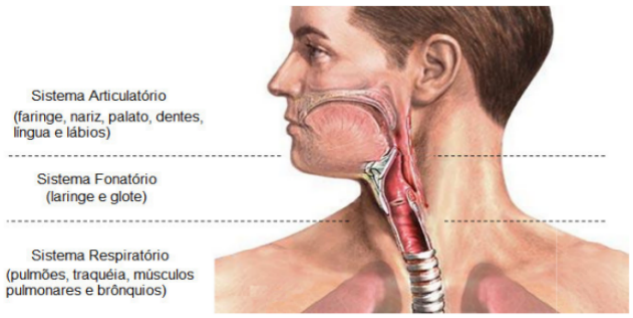

Fig. 2.1: Os Sistemas: Respiratório, Fonatório e Articulatório. Adaptação (BRESOLIN, 2008).

Portanto, quando se fala, o ar é puxado dos pulmões, passa pela garganta e pelas cordas vocais, sai pela boca e é produzida a voz. Ao falar, o trato vocal muda de forma, produzindo diferentes sons (LADEFOGED, 2001).

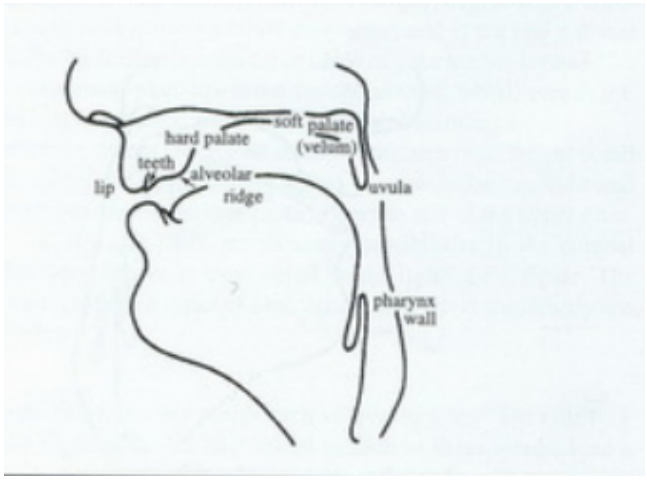

Fig. 2.2: Trato vocal parte superior. Adaptação (LADEFOGED, 2001).

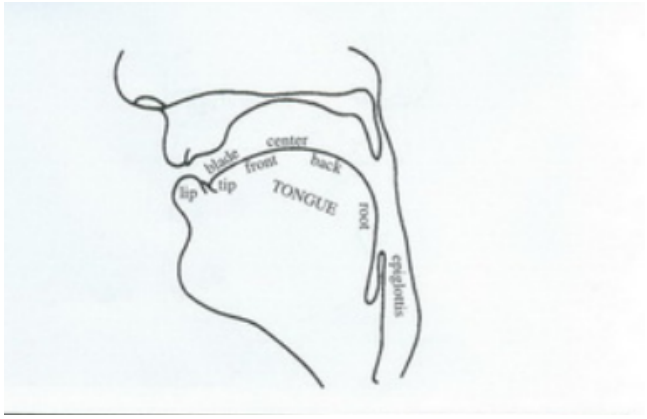

Fig. 2.3: Trato vocal parte inferior. Adaptação (LADEFOGED, 2001).

A perfeita sincronia desses três grupos de órgãos acrescidos pelas “regras de inferência” que o nosso cérebro fornece, conseguimos produzir uma voz entendível por nós que denominamos de fala.

O conjunto limitado de sons que conseguimos produzir é classificado em dois tipos de sons. Os sons sonoros (ou vozeados) que representam o vibrar das cordas e os sons surdos (ou não vozeados), para os quais as cordas vocais não vibram, apenas permanecem abertas.

2.1.1 Sociolinguística

Linguagem e sociedade matem uma relação estrita que vai além do aspecto mutacional. A Sociolinguística procura entender esse elo, cujo objeto é o estudo da língua falada, observada, descrita e analisada em seu contexto social (MUSSALIM; BENTES, 2001).

Segundo Lyons (2002), há uma grande variação entre a língua escrita e a fala. Essa variação dá-se por parâmetros como:

- Variação geográfica ou diatópica - Está relacionada a distribuição do espaço físico. Cada região possui suas particularidade na forma de pronunciar determinadas palavras, havendo uma distinção no plano lexical, fonético e gramatical.
- Variação social ou diastrática - Refere-se a identidade do falante que está relacionada a aspectos como: idade, sexo e classe social.

Para Mussalim (2001), esses aspectos evidenciam a forma variacionista da linguagem, tentando estabelecer padrões como por exemplo: o emprego do segmento sonoro /s/, que é mais utilizada na posição inicial quando queremos denotar plural, como em: “os meninos correu” em vez de “os meninos correram”.

Sem a análise da sociolinguística, um reconhecedor de voz independente de locução não conseguirá ter uma boa taxa de acerto, pois o mesmo será incapaz de tratar a variação da fala de cada locutor.

2.1.2 Fonética e Fonologia

A fonética e a fonologia são as áreas da linguística que estudam os sons da fala. A fonética visa o estudo do ponto de vista articulatório, verificando como os sons são produzidos pelo o aparelho fonador. Já a fonologia dedica-se ao estudo dos sistemas de sons, de sua descrição, estrutura e funcionamento (MUSSALIM; BENTES, 2001).

Com intuito de explorar os métodos para descrição, classificação e transcrição dos sons da fala, a fonética divide em quatro focos de estudos, que segundo LADEFOGED (2001) são:

- Fonética Articulatória: Descreve como a fala é produzida do ponto de vista articulatório e fisiológico.

- Fonética Auditiva: Compreende o estudo da percepção da fala.
- Fonética Acústica: Compreende o estudo das propriedades físicas dos sons da fala, a partir da sua transmissão do falante ao ouvinte.

Para Ladefoged (2001), a fonética articulatória classifica os segmentos consonantais com base no posicionamento entre os articuladores ativos e passivos, dividindo em oito categorias:

- Bilabial - Essa consoante é formado pela obstrução da passagem do ar que resulta no movimento do lábio contra o outro, sendo que o lábio inferior é o articulador ativo e o lábio superior é o articulador passivo. Exemplo: /p/, /m/, /b/.
- Labiodental - O articulador ativo é o lábio inferior e o passivo são os dentes incisivos superiores. Exemplos: /f/, /v/.
- Dental - Nessa consoante o articulador ativo é a língua (ápice ou lâmina), e seus articuladores passivos são os dentes incisivos superiores. Exemplo: data.
- Alveolar - São as consoante cujo som é articulado no encontro da ponta da língua com os alvéolos dentários. O articulador ativo é a língua (ápice ou lâmina) e o passivo são os alvéolos. Exemplo: lata.
- Alveopalatal - Esta consoante também é chamada de pós-velares. Onde o articulador ativo é a parte anterior da língua e o passivo é a parte medial do palato duro (céu da boca). Exemplos: tia, dia.
- Palatal - A sua pronúncia é formado pela aproximação ou o contato do dorso da língua com o palato duro. O articulador ativo é a parte média da língua e o passivo é a parte final do palato duro. Exemplo: palha.
- Velar - É formado pela aproximação ou o contato da língua com o palato mole (véu palatino). O articulador ativo é a parte posterior da língua e o passivo é o palato mole. Exemplo: gata, rata.
- Glotal - Em sua pronúncia o ponto de articulação é o glote que comporta-se como articuladores. Exemplo: a palavra escarrar, pronunciando o /r/ ao mesmo tempo.

Para Lyons (2002), a principal diferença entre a articulação das vogais e das consoantes está no fato de que para identificar a vogal precisa-se olhar a totalidade da cavidade oral, pois no mesmo há ausência de obstrução à passagem do ar pela boca.

O mesmo autor revela que existem diversas maneiras de classificar a formação das vogais, dentre elas destacam-se:

- Vogal Fechada (ou alta) - Esse som é formado com os maxilares próximos um dos outros. Exemplo: [i], [u].
- Vogal Aberta (ou baixa) - Os maxilares estão distantes entre si, devido a abertura da boca para produzir esse som. Exemplo: [a].
- Vogal Anterior - Produzida pela elevação da língua. Exemplo: [i].
- Vogal Posterior - Produzida pela retração da língua. Exemplo: [u].
- Vogal Arredondada - Som produzido pelo arredondamento dos lábios. Exemplo: [o].
- Vogal Não Arredondada - Não há arredondamento dos lábios. Exemplo: [i].

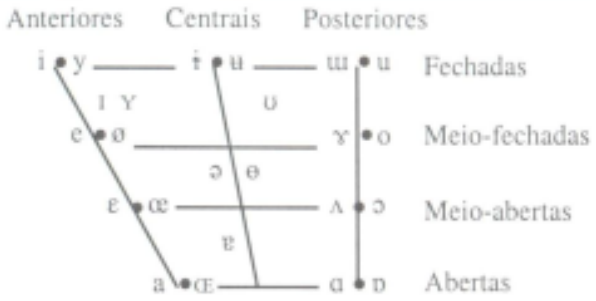

Fig. 2.4: Classificação das vogais. (MUSSALIM, 2001).

A forma como o ar é expelido pelo trato vocal, dá origem a várias propriedades sonoras de fonemas, denominados traços distintivos.

Traços distintivos são propriedades mínimas, de caráter acústico, como “nasalidade”, “sonoridade”, “labialidade”,“coronalidade”, que, de forma concorrente, constituem os sons das línguas (BISOL, 2005).

Chomsky e Halle (1968) apud Bisol (2005), descrevem os traços distintivos em: traço de classe principal, traço de cavidade, traço de modo de articulação e traço de fontes.

O traço de classe principal classifica-se em: sons soantes, silábico e consonantal. Nas soantes os sons são produzidos com uma configuração de trato vocal na qual é possível a sonorização espontânea. Os silábicos são os segmentos que constituem pico de sílaba. E os consonantais são produzidos com uma obstrução radial na cavidade oral.

	Soante	Consoante	Silábico
Vogais	+	-	-
Líquidas e Nasais não-silábicas	+	+	-
Líquidas e Nasais silábicas	+	+	+
Glides	+	-	-
Obstruintes	-	+	-

Tab. 2.1: Traços da classe principal.

Já no traço de cavidade, Chomsky e Halle (1968) apud Bisol (2005) classificam os sons em:

- Coronal - produzidos com a lâmina na língua elevada acima da posição neutra.
- Anterior - produzidos com uma obstrução localizada na frente da região palato-alveolar da boca.
- Traços relacionados com o corpo da língua que dividem em:

    1. Alto - produzidos com a elevação do corpo da língua acima da posição neutra.
    2. Baixos - produzidos com o abaixamento do corpo da língua abaixo da posição neutra.
    3. Posterior - produzidos com a retração do corpo da língua a partir da posição neutra.
    4. Arrendondado - produzidos com o estreitamento do orifício do lábio.

- Traços de aberturas secundárias:

    1. Nasal - produzidos com o abaixamento do véu palatino, permitindo o escape de ar através do nariz.
    2. Lateral - produzidos com a elevação da lâmina da língua e o abaixamento do centro da língua, permitindo o escape do ar por um lado ou por ambos os lados.

No traço de modo de articulação, têm-se: sons contínuo, a constrição primária do trato vocal não está estreitado a ponto de bloquear a passagem de fluxo de ar; sons metástase instantânea, afetam sons produzidos com fechamento no trato vocal e especifica a forma de soltura do ar; sons tenso, que são produzidos com uma ação que envolve considerável esforço muscular.

Já o traço de fonte divide-se em: sonoro e estridente. O sonoro são produzidos com vibração das cordas vocais e os estridentes que são marcados acusticamente por um ruído estridente, em virtude de uma obstrução na cavidade oral que permite a passagem do ar através de uma constrição estreita.

2.2 Modelo de voz

Um reconhecedor de voz só é capaz de identificar as palavras pronunciadas, se estiver modelada acusticamente. Esta modelagem, comumente chamada de modelo de voz, é dividido em: modelo acústico, modelo linguístico e modelo gramatical (MORAIS, 1997).

O modelo acústico é construído através de um procedimento hierárquico, utilizando como base a decomposição de sentenças em palavras e estas em unidades sub-léxicas menores.

Alguns autores dividem a construção do modelo acústico em três etapas:

- Construção do modelo estatístico para cada unidade sub-léxica.
- Construção do modelo estatístico para cada uma das palavras do léxico (nessa fase o dicionário de pronúncia já está em posse).
- Construção do modelo de sentença a partir dos modelos de palavras.

O modelo linguístico 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 (YNOGUTI, 1999).

Este modelo pode ser utilizado para determinar a probabilidade de uma palavra, em uma sentença, dadas todas as palavras que a precedem.

> 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 (YNOGUTI, 1999).

O modelo gramatical, além de possuir um dicionário de pronúncia, descreve a probabilidade de ocorrência de cada tipo de construção gramatical. Quando encontramos fonemas do tipo “foi a três anos” sugerem termos como “foia” e “trêzanos”, que não estão no dicionário de vocábulos

conhecidos, logo, são descartados. Por outro lado, ao ouvir a palavra “comunicação”, o reconhecedor de voz fará uma consulta a esse modelo, decidindo se ouviu “comunica acção” ou um único termo. O modelo gramatical vai indicar qual é a construção mais plausível (YNOGUTI, 1999).

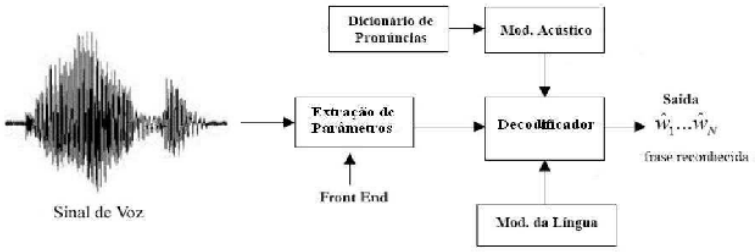

Fig. 2.5: Etapas do modelo de voz para o reconhecimento da fala. (BRESOLIN, 2008).

A figura acima mostra as etapas do modelo de voz para o reconhecimento da fala. Primeiro temos o sinal da voz com todos os ruídos, em seguida, extraímos os ruídos deixando a voz a mais limpa possível, logo após, o formato da onda é enviada para o decodificador que receberá os parâmetros do modelo acústico (já incluído o gramatical) e o modelo linguístico, gerando assim um reconhecimento da palavra dita pelo usuário.

2.2.1 Unidades Menores que a palavra

Para BISPO (1997), um reconhecedor de voz pode ser caracterizado, entre outros fatores, pela unidade fonética utilizada. Um sistema 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, trifones e difones são alternativas bastante razoáveis, pois agora é necessário ter vários exemplos de cada sub-unidade e não vários exemplos de cada palavra.

Os fonemas são as menores unidades fonéticas da língua que estabelecem papel distintivos. Por exemplo: o /p/ e /b/ representam fonemas diferentes, pois diferenciam palavras como “basta” e “pasta” (SIMOES, 1999).

Os difones são unidades que englobam somente uma transição entre um fone e outro e tem em

seus limites os pontos de máxima estabilidade espectral, ou de mínima dinâmica articulatória. Englobando apenas parte dos vários efeitos coarticulatório da língua falada, que geralmente afetam um fone inteiro (SOLEWICZ; MORAES; ALCAIM, 1994).

Essa desvantagem dos difones levou a criação dos trifones que engloba um fone inteiro e suas transições à direita e à esquerda e que constituem um complemento aos difones, podendo solucionar os efeitos dinâmicos citados. A associação entre difones e trifones para cobrir efeitos contextuais, deu origem a técnica chamada de polifones (LATSCH, 2005).

Existem dois critérios importantes que são analisados durante a escolha dessas unidades: consistência e treinabilidade.

> Consistência - a unidade deve ter características similares em sentenças diferentes. É importante porque permite uma discriminação efetiva entre unidades distintas.
> Treinabilidade - devem existir amostras suficientes para o treinamento e a criação de um modelo com bom desempenho nos testes. Sua importância reside no fato de os modelos atualmente usados no reconhecimento exigirem grandes quantidades de dados de treinamento (ALENCAR, 2005).

Ao longo dos anos, vários trabalhos foram propostos na tentativa de explicar qual a melhor unidade fonética a utilizar. Dentre eles destacam-se:

- Malbos et al (MALBOS; BAUDRY; MONTRESOR, 1994) foram um dos primeiros trabalhos que utilizou Wavelet em RAF, os autores escolheram consoantes oclusivas (/p/, /k/, /t/, /b/, /g/ e /d/) aplicadas a língua francesa.
- Marchesi et al (MARCHESI; LIPPMANN; NOHAMA, 1996) fizeram um estudo de reconhecimento das vogais orais do português brasileiro, utilizando as frequências fundamentais como descritores.
- Alcaim (A. ALCAIM, 2001), utilizou as sílabas como unidade. Porém, alguns autores alertaram que esse método só se apresentava atraente para o desenvolvimento de um RAF com poucos padrões.

- Deshmukh et al (DESHMUKH; ESPY-WILSON; JUNEJA, 2002) utilizaram os parâmetros acústico-fonético no reconhecimento de voz.

Os trabalhos revelam que a melhor estratégia no desenvolvimento de um RAF é a utilização de unidades menores que a palavra. A escolha de qual utilizar depende do propósito do RAF e em qual língua ele será moldado. Segundo Young (1996), a melhor estratégia para o Inglês é a utilização de trifones. Mesmo com a dificuldade de treinabilidade, ocasionado pelo idioma em ter uma difícil separação silábica.

2.2.2 Modelo Oculto de Markov

Quase a totalidade dos sistemas que representam o estado-da-arte em reconhecimento de fala constroem seus modelos acústicos baseados no modelo oculto de Markov, cujo sucesso se dá pela capacidade de modelar tanto a variabilidade acústica como temporais do sinal da fala e também por permitir a construção hierárquica dos modelos acústicos (MORAIS, 1997).

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). Markov baseia-se na suposição de que um processo continuo pode ser aproximado por uma sucessão de curtos estados estacionários e modela uma sequência de vetores acústicos como um processo estacionário por partes (YNOGUTI, 1999).

O HMM (Modelo oculto de Markov) é definido como um par de processos estocásticos, representam processos não observáveis e observáveis.

Os estados não observáveis (oculto) representam algumas características físicas do problema. Suas transições são definidas de acordo com uma matriz de probabilidade gerada por um algorítimo específico baseado em eventos aleatórios (CHING; NG, 2006).

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, 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 (YNOGUTI, 1999).

A figura 2.6, mostra um exemplo de um modelo left-right com 3 estados.

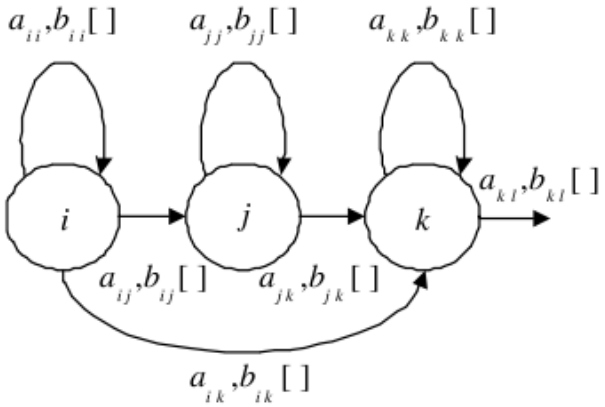

Fig. 2.6: Modelo HMM left- right de 3 estados. (YNOGUTI, 1999).

Segundo Morais (1997), em geral, três problemas são levantados quando utiliza HMM para modelar sequência:

- Avaliação - Busca a solução da dificuldade de definir qual a verossimilhança de um HMM gerar uma dada sequência de observações. Geralmente, esse problema ocorre no reconhecimento de palavras isoladas, quando tentamos verificar qual delas apresentam a maior verossimilhança de ter gerado a palavra a ser reconhecida.
- Decodificação - Trilha os caminhos (sequência de estado) no reconhecimento da palavra com maior verossimilhança ao longo de uma sequência de vetores acústicos.
- Estimativa de parâmetros ou treinamento - Estipula como e quando treinar a estrutura do reconhecedor da voz, para que esse modelo maximize a verossimilhança da geração de sequências de observação.

Segundo Rabiner e Juang (1993), um HMM é definido por:

$$A = \left\{ a_{ij} | a_{ij} = P(q^{n+1} = j | q^n = i) = P(q_j^{n+1} | q_j^n) \right\} \tag{2.1}$$

$$B = \left\{ b_j(x_i) | b_j(x_i) = P(x_i | q = j) = P(x_i | q_i) \right\} \tag{2.2}$$

$$\pi = \left\{ \pi_i | \pi_i = P(q^1 = i) = P(q_i^1) \right\}. \tag{2.3}$$

A expressão (2.1) distribui a probabilidade de transição de um estado i para o estado j, sempre possuindo a mesma probabilidade. Na expressão (2.2), para cada par $P(xi|qi)$ há uma distribuição de probabilidade de retorno correspondente no caso de um HMM discreto, ou uma função distribuição de probabilidade para um HMM contínuo. O termo $b_j(x_i)$ refere-se a probabilidade (no caso discreto) ou verossimilhança (no caso contínuo) do símbolo x_i gerado pelo estado q_j. Finalmente, a expressão (2.3) refere-se a probabilidade de distribuição do primeiro estado. No modelo left-right, é assumido $\pi_1 = 1$ e $\pi_i = 0$ para cada $i \neq 1$.

Uma característica importante dos HMM's é a capacidade de dinâmica de eventos do modelo no domínio do tempo. Assim, eles podem ser implementados para reconhecimento de padrões no sinal de fala como na análise temporal, análise acústica ou ambos (JUANG; RABINER, 1991).

Vários trabalhos de reconhecimento de voz foram implementados utilizando o classificador HMM, dentre eles destacam-se:

- Saadeq et al (SAADEQ; ALI, 2010) propuseram um RAF aplicada a eletromiográfia e vibrocervigrafia, modelada com HMM temporal. Os autores obtiveram uma diminuição significativa no efeito ruído.
- Kim et al (KIM; YOUN; LEE, 2000) usaram HMM juntamente com Wavelet mãe ortogonal para reconhecer dígitos Coreanos. Os autores obtiveram uma taxa de acerto superior ao descritor MFCC (Mel-Frequência Cepestrais Coeficientes).
- Hwang (HWANG; HUANG, 1991) propôs um modelo de compartilhamento distribuído para ASR independente utilizando HMM. Os autores obtiveram uma redução de estados redundantes ocasionando uma melhor taxa de reconhecimento.

Outros trabalhos sobre reconhecimento de voz utilizando HMM são: Tolba (2009), Tan et al. (1996), Sarikaya e Gowdy (1998), Zhu e Alwan (2000), Hosom (2002), Chan, Ching e Lee (2001) e Jiang, Meng e Gao (2003).

2.3 Processamento de sinal

O processamento do sinal da fala tem como objetivo converter o sinal em parâmetros entendíveis para as etapas seguintes do reconhecedor. Para tal, esta etapa é dividida em dois blocos: Pré-processamento e obtenção dos atributos da voz (ALENCAR, 2005).

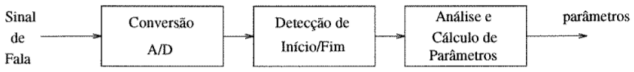

Fig. 2.7: Processamento do sinal da fala em um reconhecedor.(MARTINS, 1997).

2.3.1 Pré-processamento

A primeira etapa no pré-processamento da voz é a conversão do sinal analógico em digital (conversor A/D). Através de um transdutor que, em geral, é um microfone. Uma boa execução dessa primeira etapa, marcação de início e fim de locução bem realizada, garante excelentes taxas de reconhecimento pois há uma diminuição do efeito ruído sobre a elocução (JÚNIOR, 2009).

A amostragem de voz geralmente é efetuada entre 6k a 44kHz, sofrendo uma codificação linear de 8 ou 16 bits, satisfazendo o teorema de Nyquist. Em muitos sistemas é comum a aplicação de um filtro passa-baixas para limitar a banda de frequência do sinal. Com isto, pode-se eliminar o fenômeno conhecido como aliasing (OPPENHEIM; WILLSKY; NAWAB, 1996).

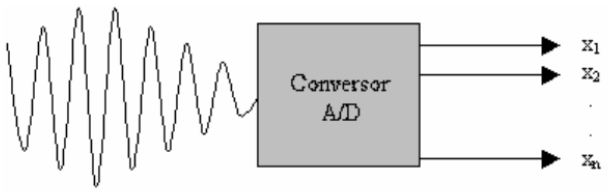

Fig. 2.8: Conversão analógico digital. (VALIATI, 2000).

A segunda etapa é atenuar os componentes de baixa frequência do sinal, prevenindo contra instabilidade numérica e minimizando o efeito do glote e do lábio. Essa etapa denomina-se de pré-ênfase (SOTOMAYOR, 2003).

Segundo Alencar (2005), a função de pré-ênfase mais utilizada é a de primeira ordem fixo, dada

por:

$$H(z) = 1 - az^{-1} \tag{2.4}$$

O valor mais comum que "a" assumi é aproximadamente 0,95. A saída da pré-ênfase denominada de s(n), está relacionada com a entrada, e(n), pela equação:

$$s(n) = e(n) - ae(n-1) \tag{2.5}$$

Para Rabiner e Juang (1993), a terceira etapa é a extração de quadros da amostra do sinal. Isso ocorre devido o sinal ser variante no tempo. Por isso, defini-se um janelamento entre 10-45 ms que é movida ao longo do sinal da voz, com ou sem superposicionamento entre quadros adjacentes.

Existem diversas formas de implementar o janelamento do sinal, a mais comum foi proposta por Oppenheim, Willsky e Nawab (1996), definida como:

$$w(n) = 0,54 - 0,46\cos(\frac{2n\pi}{N_w - 1}) \tag{2.6}$$

Sendo que *Nw* é o tamanho da janela, onde de cada quadro são extraídos os atributos dos sinais de voz. A figura 2.9 ilustra a análise do sinal de curta duração.

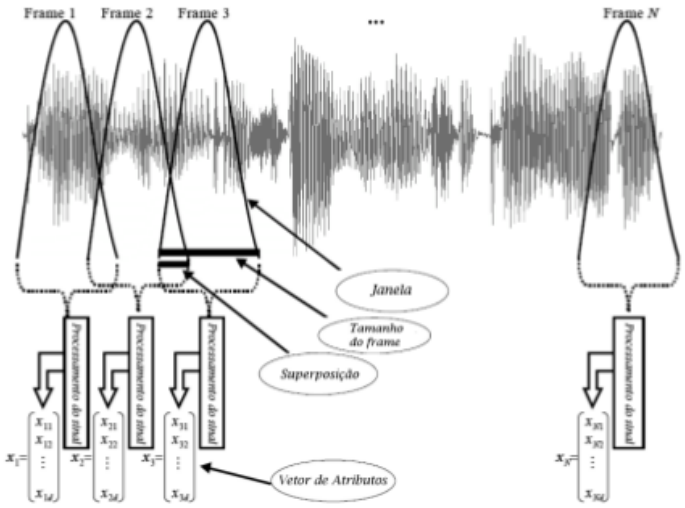

Fig. 2.9: Visão geral da análise do sinal de curta duração. (JÚNIOR, 2009).

A última etapa é chamada de endpoints (pontos terminais), que evita o processamento dos segmentos onde não há voz, antes e depois do sinal com voz, evitando carga computacional e economizando tempo, além de servir como marco de início e fim de um segmento de voz. A determinação dos endpoints deve ser feita de forma cuidadosa, pois os mínimos erros nesta estimação podem degradar o reconhecimento (SANTOS, 2001).

2.3.2 Mel-Frequência Cepestrais Coeficientes (MFCC)

O MFCC são atributos obtido da voz, chamado de descritor. Ela é a mais utilizada por pesquisadores em processamento de sinal, devido ao fato de possuírem um elevado desempenho (BRESOLIN, 2008).

> Os coeficientes Mel-cepestrais surgiram devido aos estudos na área de psicoacústica (ciência que estuda a percepção auditiva humana), que mostraram que a percepção humana das freqüências de tons puros ou de sinais de voz, não seguem uma escala linear. Isto estimulou a idéia de serem definidas freqüências subjetivas de tons puros, da seguinte forma: para cada tom com freqüência f, medida em Hz, define-se um tom subjetivo medido em uma escala que se chama escala mel. O mel, então, é uma unidade de medida da freqüência percebida de um tom (ALENCAR, 2005).

Com isso, definiu-se frequência de 1 kHz, com potência 40 dB acima do limiar mínimo de audição do ouvido humano, como 1000 mels. Graças a ponderação da escala de frequência para a escala mel, pode-se identificar a banda crítica, que são sons não individualmente identificados, dentro de certas bandas, pelo ouvido humano (CUADROS, 2007).

Para Martins (1997), o cálculo do MFCC é feito utilizando um banco de filtros espaçados na escala Mel e o cálculo do logaritmo da energia na saída de cada filtro seguido de uma transformada discreta do cosseno.

$$c(n) = \sum_{k=1}^{M} \log_{10} X(k) \cos(n(k-1/2)\pi/M) \tag{2.7}$$

Onde:

- $1 \leq$ n $\leq$ N.
- X(k) é a energia na saída do k-ésimo filtro.
- M é o número de filtros.
- N é o número de coeficientes.

A figura 2.10 mostra uma representação do processo de obtenção dos coeficientes MFCC.

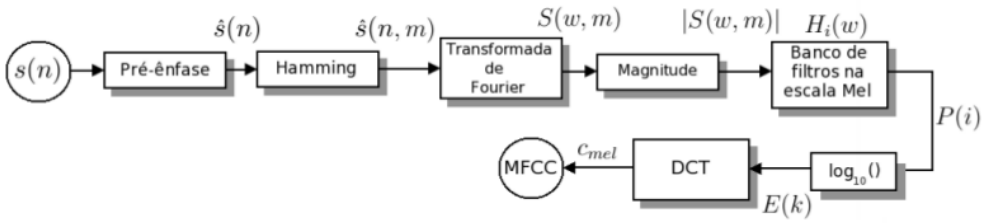

Fig. 2.10: Diagrama de fluxo para o cálculo dos MFCCs (CUADROS, 2007).

3 *Resultado e Discussão*

Este trabalho construiu um modelo de voz em português brasileiro, com base nas premissas da fonética e fonologia, utilizando S.O. (Sistema Operacional) Linux e a ferramenta livre HTK (Hidden Markov Model Toolkit).

Para validar a proposta, foi desenvolvido um programa reconhecedor de palavras isoladas, integrando ao decodificador Julius.

3.1 Preparação dos dados

Antes de construirmos o modelo de voz, precisa-se instalar o HTK. O mesmo é restrito ao uso não comercial e é encontrado no site *"http://htk.eng.cam.ac.uk/download.shtml"*. Para instalá-lo, a compilação do seu fonte é necessária observando a arquitetura do computador.

O próximo passo é a instalação do decodificador Julius. Que se encontra nos repositórios Debian e também no site *"http://julius.sourceforge.jp/"*.

Os compiladores do Python e C deverão ser instalados, pois, são necessários na execução dos programas de treinamento do HTK e dos scripts propostos para automatização das tarefas.

3.2 HTK

O HTK (Hidden Markov Model Toolkit) é definido como um conjunto portátil de ferramentas de software para construção e manipulação de HMM. Seu código fonte esta disponível em C. Esta ferramenta provê sofisticadas funcionalidades para análise da fala, treino de HMM, testes e análise de resultados. Sua vasta documentação está disponível no site *"http://htk.eng.cam.ac.uk/download.shtml"* com o nome de "HTK Book" (YOUNG, 1994).

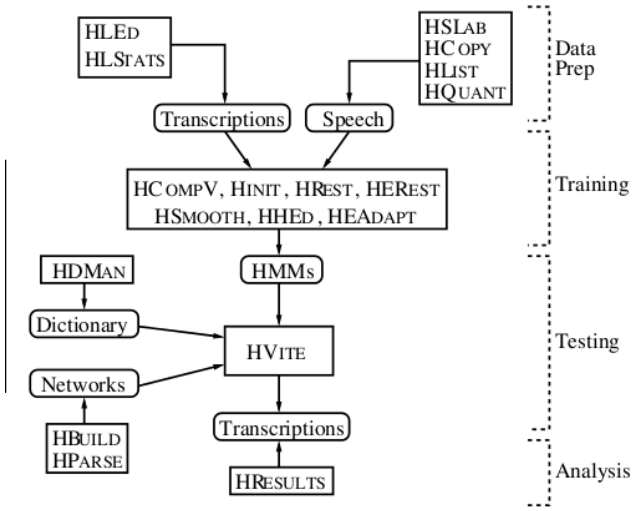

Fig. 3.1: Estrutura de ferramentas disponíveis no pacote HTK para testes de sistemas. (YOUNG, 1994).

A figura 3.1 mostra as estruturas das ferramentas do HTK. Alguns desses softwares foram utilizados para treinamento, dentre eles destacam:

- HDMan -> Usado para preparar o dicionário de pronúncia de forma automática a partir do léxico.
- HLEd -> Substitui as palavras de cada arquivo marcado para treinamento pelos fonemas correspondentes.
- HCopy -> Faz a conversão dos arquivos WAV para MFCC.
- HCompV -> Calcula um vetor de variação médio de cada estado de treinamento.
- HERest -> Usado para re-estimar parâmetros em um conjunto de HMM, usando uma versão de treinamento integrado do algoritmo Baum-Welch.
- HHEd -> Manipulador de HMM. Sua função principal é aplicar uma sequência de operações, em cada estado, no conjunto do HMM .
- HVite -> Software para treinamento semelhante ao HLEd, porém observa-se a pronúncia da mesma palavra várias vezes, escolhendo a que mais coincide com os dados acústico.

3.3 Construção do modelo de voz

A construção da gramática faz-se necessária, pois define regras para reconhecimento. O formato das regras é baseado na BNF(Forma Normal de Backus), seguindo a derivação:

Símbolo:expressão

Onde:

- Símbolo -> É não terminal;
- Expressão -> Sequência de símbolos que podem ser terminal ou não-terminal.

Um símbolo terminal é aquele que representa um valor constante, enquanto que não-terminal pode ser expresso em termo de outros símbolos (HOPCROFT; ULLMAN; MOTAWANI, 2008).

A imagem 3.2 mostra as regras utilizadas.

```
S : INICIO_SILENCIO COMANDO FIM_SILENCIO
COMANDO : CHAMADA NOME
COMANDO : DIA TEMPO
```

Fig. 3.2: Regras da gramática utilizada na criação do modelo de voz.

O símbolo inicial, aqui marcado por “S”, possui marcação de silêncio tanto no início do comando quanto no fim. Essa marcação é responsável por definir o início e o fim do comando.

O próximo passo é a definição de cada comando (palavras) definido na gramática. Para tal, foi criado um arquivo com extensão .voca, pois, para fins de treinamento, é a extensão reconhecida pelo HTK.

A imagem 3.3 revela os comandos definidos.

Após definir as duas vertentes da gramática, é hora de gerar os autômatos finitos. Isso é feito unindo a definição da gramática com o arquivo .voca, explicitando em qual estado do autômato está cada comando. Porém, para que o decodificador Julius reconheça estes estados, foi utilizado o script mkdfa.pl que gera um arquivo .dfa cujo conteúdo é a representação dos autômatos.

A figura 3.4 e 3.5 mostram os arquivos de autômatos gerados. A figura 3.6 é a saída do script mkdfa.pl.

```
% INICIO_SILENCIO
<s>     sil
% FIM_SILENCIO
</s>    sil

% CHAMADA
COMPUTA         k o~  p u t a
COMPUTADOR      K O~  p u t a d o r

% NOME
LIGAR           l i g a r
DESLIGAR        d e s l i g a r
ESCUTE          e s k u t e
NAVEGADOR       n a v e g a d o r
EDITOR          e dZ  i t o r
TERMINAL        t e R m i~  n a w
AGENDA          a Z e~  d a
MUSICA          m u z i k a

% DIA
SEGUNDA         s e g u~  d a
TERCA           t e r s a
QUARTA          k w a r t a
QUINTA          k i~  t a
SEXTA           s e s t a
SABADO          s a b a d o
DOMINGO         d o m i~  g u

% TEMPO
QUENTE          k e~  tS i
FRIO            f r i u
CHUVOSO         S u v o z u
SOLARADO        s o l a r a d u
```

Fig. 3.3: Comandos definidos para a criação do modelo de voz.

```
0       INICIO_SILENCIO
1       FIM_SILENCIO
2       CHAMADA
3       NOME
4       DIA
5       TEMPO
```

Fig. 3.4: Definição dos estados dos autômatos.

```
0       [<s>]   sil
1       [</s>]  sil
2       [COMPUTA]       k o~ p u t a
2       [COMPUTADOR]    K O~ p u t a d o r
3       [LIGAR]         l i g a r
3       [DESLIGAR]      d e s l i g a r
3       [ESCUTE]        e s k u t e
3       [NAVEGADOR]     n a v e g a d o r
3       [EDITOR]        e dZ i t o r
3       [TERMINAL]      t e R m i~ n a w
3       [AGENDA]        a Z e~ d a
3       [MUSICA]        m u z i k a
4       [SEGUNDA]       s e g u~ d a
4       [TERCA]         t e r s a
4       [QUARTA]        k w a r t a
4       [QUINTA]        k i~ t a
4       [SEXTA]         s e s t a
4       [SABADO]        s a b a d o
4       [DOMINGO]       d o m i~ g u
5       [QUENTE]        k e~ tS i
5       [FRIO]          f r i u
5       [CHUVOSO]       S u v o z u
5       [SOLARADO]      s o l a r a d u
```

Fig. 3.5: Geração dos autômatos.

```
0 1 1 0 0
1 3 2 0 0
1 5 3 0 0
2 2 4 0 0
3 4 4 0 0
4 0 5 0 0
5 -1 -1 1 0
```

Fig. 3.6: Representação dos autômatos gerado pelo script mkdfa.pl.

A próxima etapa é a construção do dicionário de pronúncia. Para tal, foram realizados os seguintes passos:

1. Criação da lista de palavras que será gravada, adicionando as palavras sent-start e sent-end necessárias para que o decodificador Julius faça o processamento do modelo acústico. Ver figura 3.7.
2. Arquivos de instruções indicando o local e qual sequência das palavras que foram gravadas em determinado arquivo. A figura 3.8 exemplifica esse passo.

Ao término desses passos, a integração dos arquivos foi feita utilizando o programa HDMan da ferramenta HTK, produzindo um dicionário de palavras, uma lista e uma estatística dos fonemas. Para gerar estes resultados, é necessária a criação de um arquivo padrão denominado de global.ded que insere uma pausa entre cada palavra. As figuras 3.9, 3.10, 3.11 e 3.12 mostram a saída dos comando.

O uso do software HDMan no auxílio da construção do dicionário de pronúncia é de caráter opcional, seu resultado poderá ser produzido manualmente sem perda de integração. Nesse trabalho, os dois métodos foram testados, apresentando a mesma velocidade e taxa de reconhecimento.

As vozes foram gravadas com o auxílio da ferramenta Audacity. Esse procedimento foi realizado com 6 adultos do sexo masculino e 4 adultos do sexo feminino com idade entre 18 a 50 anos. A frequência estabelecida foi 48000HZ e a amplitude da onda variou entre 0,5 e -0,5, pois, entre a amplitude 1 e -1 a voz apresentava distorção. O formato estabelecido foi WAV de 16 bits que é o padrão do programa HCopy, responsável por converter o WAV para MFCC. A figura 3.13 mostra a gravação das palavras: chamada, computa, nome, computa, nome, nome, nome e chamada.

A conversão do áudio para o MFCC é necessária, pois, o HTK apresenta baixa eficiência com

```
AGENDA
CHAMADA
CHUVOSO
COMPUTA
COMPUTADOR
DESLIGAR
DIA
DOMINGO
EDITOR
ESCUTE
FRIO
LIGAR
MUSICA
NAVEGADOR
NOME
QUARTA
QUENTE
QUINTA
SABADO
SEGUNDA
SENT-END
SENT-START
SEXTA
SOLARADO
TEMPO
TERCA
TERMINAL
```

Fig. 3.7: Lista de palavras gravadas.

```
*/arquivo1 CHAMADA COMPUTA COMPUTA COMPUTA COMPUTA COMPUTA COMPUTA COMPUTA
*/arquivo2 CHAMADA COMPUTA NOME COMPUTA NOME NOME NOME CHAMADA
*/arquivo3 NOME CHAMADA COMPUTA DESLIGAR DESLIGAR DESLIGAR DESLIGAR DESLIGAR
*/arquivo4 LIGAR LIGAR DESLIGAR LIGAR DESLIGAR LIGAR DESLIGAR MUSICA
*/arquivo5 ESCUTE ESCUTE ESCUTE ESCUTE ESCUTE ESCUTE MUSICA COMPUTA
*/arquivo6 DESLIGAR LIGAR ESCUTE LIGAR DESLIGAR ESCUTE DESLIGAR TERMINAL
*/arquivo7 LIGAR DESLIGAR ESCUTE COMPUTA DESLIGAR LIGAR ESCUTE DESLIGAR
*/arquivo8 COMPUTADOR COMPUTADOR ESCUTE AGENDA ESCUTE AGENDA AGENDA AGENDA
*/arquivo9 COMPUTADOR ESCUTE AGENDA LIGAR DESLIGAR AGENDA AGENDA AGENDA
*/arquivo10 COMPUTA COMPUTADOR EDITOR EDITOR AGENDA DESLIGAR EDITOR EDITOR
*/arquivo11 NOME AGENDA EDITOR TERMINAL MUSICA MUSICA MUSICA TERMINAL
*/arquivo12 NOME TERMINAL NOME MUSICA TERMINAL AGENDA NOME MUSICA
*/arquivo13 MUSICA MUSICA MUSICA MUSICA TERMINAL MUSICA TERMINAL NOME
*/arquivo14 NOME NAVEGADOR NAVEGADOR NAVEGADOR NOME NAVEGADOR NAVEGADOR NAVEGADOR
*/arquivo15 COMPUTA COMPUTADOR NAVEGADOR ESCUTE NAVEGADOR ESCUTE COMPUTA COMPUTADOR
*/arquivo16 MUSICA ESCUTE COMPUTADOR COMPUTA NAVEGADOR DESLIGAR TERMINAL TERMINAL
*/arquivo17 DIA SEGUNDA TERCA QUARTA QUINTA SEXTA SABADO DOMINGO
*/arquivo18 DIA TERCA SEXTA DOMINGO QUARTA SABADO QUINTA SEGUNDA
*/arquivo19 DIA QUARTA QUINTA SEXTA SEGUNDA TERCA SABADO DOMINGO
*/arquivo20 DIA QUINTA SEGUNDA SEXTA DOMINGO SABADO TERCA QUARTA
*/arquivo21 TEMPO SOLARADO QUENTE FRIO CHUVOSO TEMPO SOLARADO FRIO
*/arquivo22 TEMPO QUENTE FRIO QUENTE FRIO QUENTE QUENTE FRIO
*/arquivo23 SOLARADO SOLARADO SOLARADO TEMPO CHUVOSO CHUVOSO CHUVOSO CHUVOSO
*/arquivo24 DIA TEMPO CHAMADA COMPUTADOR CHAMADA SOLARADO FRIO QUENTE
*/arquivo25 CHAMADA NOME QUARTA DIA TEMPO FRIO QUENTE SOLARADO
*/arquivo26 TEMPO CHUVOSO FRIO QUENTE SOLARADO DIA MUSICA DOMINGO
*/arquivo27 CHAMADA TEMPO DIA SEGUNDA CHUVOSO NAVEGADOR TERMINAL LIGAR
*/arquivo28 TEMPO CHAMADA DESLIGAR CHUVOSO SOLARADO FRIO QUENTE SABADO
*/arquivo29 TERCA NOME LIGAR DESLIGAR CHUVOSO FRIO QUENTE CHAMADA
*/arquivo30 AGENDA AGENDA SOLARADO CHUVOSO TEMPO NAVEGADOR QUARTA FRIO
*/arquivo31 LIGAR DESLIGAR ESCUTE NAVEGADOR COMPUTA FRIO QUINTA MUSICA
*/arquivo32 NOME DESLIGAR TEMPO TERMINAL MUSICA SABADO AGENDA QUENTE
```

Fig. 3.8: Arquivo de instruções.

```
a
Z
e~
d
sp
x
m
S
u
v
o
z
k
o~
p
t
K
r
e
s
l
i
g
i~
dZ
f
n
w
tS
b
u~
sil
```

Fig. 3.9: Lista de fonemas.

```
New Phone Usage Counts
----------------------
 1. a     :    24
 2. Z     :     1
 3. e~    :     3
 4. d     :    10
 5. sp    :    54
 6. x     :     1
 7. m     :     5
 8. S     :     1
 9. u     :     9
10. v     :     2
11. o     :     9
12. z     :     2
13. k     :     6
14. o~    :     2
15. p     :     3
16. t     :    10
17. K     :     1
18. r     :    10
19. e     :    10
20. s     :    10
21. l     :     5
22. i     :     9
23. g     :     5
24. i~    :     3
25. dZ    :     1
26. f     :     1
27. n     :     3
28. w     :     2
29. tS    :     1
30. b     :     1
31. u~    :     1
```

Fig. 3.10: Estatística dos fonemas.

```
AGENDA          [AGENDA]        a Z e~ d a sp
CHAMADA         [CHAMADA]       x a m a d a sp
CHUVOSO         [CHUVOSO]       S u v o z u sp
COMPUTA         [COMPUTA]       k o~ p u t a sp
COMPUTADOR      [COMPUTADOR]    K o~ p u t a d o r sp
DESLIGAR        [DESLIGAR]      d e s l i g a r sp
DIA             [DIA]           d i a sp
DOMINGO         [DOMINGO]       d o m i~ g u sp
EDITOR          [EDITOR]        e dZ i t o r sp
ESCUTE          [ESCUTE]        e s k u t e sp
FRIO            [FRIO]          f r i u sp
LIGAR           [LIGAR]         l i g a r sp
MUSICA          [MUSICA]        m u z i k a sp
NAVEGADOR       [NAVEGADOR]     n a v e g a d o r sp
NOME            [NOME]          n o m e sp
QUARTA          [QUARTA]        k w a r t a sp
QUENTE          [QUENTE]        k e~ tS i sp
QUINTA          [QUINTA]        k i~ t a sp
SABADO          [SABADO]        s a b a d o sp
SEGUNDA         [SEGUNDA]       s e g u~ d a sp
SENT-END        []              s i l
SENT-START      []              s i l
SEXTA           [SEXTA]         s e s t a sp
SOLARADO        [SOLARADO]      s o l a r a d u sp
TEMPO           [TEMPO]         t e~ p o sp
TERCA           [TERCA]         t e r s a sp
TERMINAL        [TERMINAL]      t e r m i~ n a w sp
```

Fig. 3.11: Dicionário.

```
AS sp
RS cmu
MP sil sil sp
```

Fig. 3.12: Global.ded, arquivo padrão do HDMan.

Fig. 3.13: Gravação da voz com o software Audacity.

arquivos WAV. O HCopy foi o software utilizado para essa função, apresentando duas opções:

1. Executar o comando HCopy para cada arquivo de áudio.
2. Criar um arquivo com extensão “scp” contendo uma lista de todos os arquivos de áudio.

As duas opções foram testadas neste trabalho apresentando o mesmo desempenho. A figura 3.14 mostra o arquivo contendo a localização do áudio que será convertido.

```
../acustico/arquivo1.wav ../acustico/arquivo1.mfc
../acustico/arquivo2.wav ../acustico/arquivo2.mfc
../acustico/arquivo3.wav ../acustico/arquivo3.mfc
../acustico/arquivo4.wav ../acustico/arquivo4.mfc
../acustico/arquivo5.wav ../acustico/arquivo5.mfc
../acustico/arquivo6.wav ../acustico/arquivo6.mfc
../acustico/arquivo7.wav ../acustico/arquivo7.mfc
../acustico/arquivo8.wav ../acustico/arquivo8.mfc
../acustico/arquivo9.wav ../acustico/arquivo9.mfc
../acustico/arquivo10.wav ../acustico/arquivo10.mfc
../acustico/arquivo11.wav ../acustico/arquivo11.mfc
../acustico/arquivo12.wav ../acustico/arquivo12.mfc
../acustico/arquivo13.wav ../acustico/arquivo13.mfc
../acustico/arquivo14.wav ../acustico/arquivo14.mfc
../acustico/arquivo15.wav ../acustico/arquivo15.mfc
../acustico/arquivo16.wav ../acustico/arquivo16.mfc
../acustico/arquivo17.wav ../acustico/arquivo17.mfc
../acustico/arquivo18.wav ../acustico/arquivo18.mfc
../acustico/arquivo19.wav ../acustico/arquivo19.mfc
../acustico/arquivo20.wav ../acustico/arquivo20.mfc
../acustico/arquivo21.wav ../acustico/arquivo21.mfc
../acustico/arquivo22.wav ../acustico/arquivo22.mfc
../acustico/arquivo23.wav ../acustico/arquivo23.mfc
../acustico/arquivo24.wav ../acustico/arquivo24.mfc
../acustico/arquivo25.wav ../acustico/arquivo25.mfc
../acustico/arquivo26.wav ../acustico/arquivo26.mfc
../acustico/arquivo27.wav ../acustico/arquivo27.mfc
../acustico/arquivo28.wav ../acustico/arquivo28.mfc
../acustico/arquivo29.wav ../acustico/arquivo29.mfc
../acustico/arquivo30.wav ../acustico/arquivo30.mfc
../acustico/arquivo31.wav ../acustico/arquivo31.mfc
../acustico/arquivo32.wav ../acustico/arquivo32.mfc
```

Fig. 3.14: Áudio marcado para conversão.

3.4 Treinamento do modelo acústico

O primeiro passo para o treinamento, foi a criação de dois arquivos contendo os fonemas de cada palavra com suas respectivas localização de áudio. A diferença desse arquivo, está no fato que o primeiro não possui marcação de pausa entre as palavras, enquanto que no segundo há essa marcação. Estes arquivos fazem-se necessários, devido a exigência de treinamento da ferramenta HTK.

Para a criação desses arquivos, foi utilizada a ferramenta HLEd. A figura 3.15 e 3.16 mostram trechos dos fonemas criados para treinamento.

O próximo passo é a criação dos monofones. De acordo ao HTK Book, para criar os monofones, é necessária a criação de um arquivo de configuração, que possua definição de um HMM com no

```
"*/arquivo1.lab"
sil
x
a
m
a
d
a
k
o~
p
u
t
a
```

Fig. 3.15: Fonema para treinamento sem marcação de pausa.

```
"*/arquivo1.lab"
sil
x
a
m
a
d
a
sp
k
o~
p
u
t
a
sp
```

Fig. 3.16: Fonema para treinamento com marcação de pausa.

mínimo 3 estados e vetor de tamanho 39. O tamanho é calculado: vetor estático (MFCC = 13) + coeficiente delta (13) + coeficiente de aceleração (13). A figura 3.17 mostra essa configuração.

```
~o <VecSize> 25 <MFCC_0_D_N_Z>
~h "proto"
<BeginHMM>
  <NumStates> 5
  <State> 2
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <State> 4
    <Mean> 25
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 25
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <TransP> 5
  0.0 1.0 0.0 0.0 0.0
  0.0 0.6 0.4 0.0 0.0
  0.0 0.0 0.6 0.4 0.0
  0.0 0.0 0.0 0.7 0.3
  0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Fig. 3.17: Arquivo de configuração para criação de um HMM.

Em seguida, é utilizado o software HCompV, que tem como entrada a configuração mencionada, o HERest e o HHED. O primeiro cria um conjunto de HMM, o segundo re-estima para aumentar a taxa de reconhecimento e o terceiro "amarra" o silêncio (sil) com a pausa (sp) para efetivar o reconhecimento. As figura 3.18 e 3.19 mostram, respectivamente, a criação do conjunto HMM e sua re-estimação.

```
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 25
 -1.762419e-07 1.006156e-07 -1.455492e-08 5.563903e-08 -1.414202e-08 6.122616e-08 3.482171e-08 -3.767764e-09 -1.452051e-08 -1.799580e-08 -3.045179e-09 -3.408192e-08
5.459790e-04 1.520958e-03 -5.097839e-04 -3.301320e-03 -8.957308e-04 2.882648e-03 3.300986e-03 8.771736e-04 -1.816843e-03 -3.312509e-03 -6.008132e-04 9.659103e-04
1.220119e-04
<VARIANCE> 25
 4.888231e+01 5.055955e+01 4.227111e+01 5.924963e+01 5.040355e+01 3.593556e+01 4.213013e+01 3.193390e+01 2.497008e+01 2.051501e+01 2.155564e+01 2.510994e+01 1.201700e
+00 2.015445e+00 1.348279e+00 1.589908e+00 1.876995e+00 1.810325e+00 1.991898e+00 1.682100e+00 1.658763e+00 1.468302e+00 1.457561e+00 1.262385e+00 1.134145e+00
<GCONST> 9.452953e+01
<STATE> 3
<MEAN> 25
 -1.762419e-07 1.006156e-07 -1.455492e-08 5.563903e-08 -1.414202e-08 6.122616e-08 3.482171e-08 -3.767764e-09 -1.452051e-08 -1.799580e-08 -3.045179e-09 -3.408192e-08
5.459790e-04 1.520958e-03 -5.097839e-04 -3.301320e-03 -8.957308e-04 2.882648e-03 3.300986e-03 8.771736e-04 -1.816843e-03 -3.312509e-03 -6.008132e-04 9.659103e-04
1.220119e-04
<VARIANCE> 25
 4.888231e+01 5.055955e+01 4.227111e+01 5.924963e+01 5.040355e+01 3.593556e+01 4.213013e+01 3.193390e+01 2.497008e+01 2.051501e+01 2.155564e+01 2.510994e+01 1.201700e
+00 2.015445e+00 1.348279e+00 1.589908e+00 1.876995e+00 1.810325e+00 1.991898e+00 1.682100e+00 1.658763e+00 1.468302e+00 1.457561e+00 1.262385e+00 1.134145e+00
<GCONST> 9.452953e+01
<STATE> 4
<MEAN> 25
 -1.762419e-07 1.006156e-07 -1.455492e-08 5.563903e-08 -1.414202e-08 6.122616e-08 3.482171e-08 -3.767764e-09 -1.452051e-08 -1.799580e-08 -3.045179e-09 -3.408192e-08
5.459790e-04 1.520958e-03 -5.097839e-04 -3.301320e-03 -8.957308e-04 2.882648e-03 3.300986e-03 8.771736e-04 -1.816843e-03 -3.312509e-03 -6.008132e-04 9.659103e-04
1.220119e-04
<VARIANCE> 25
 4.888231e+01 5.055955e+01 4.227111e+01 5.924963e+01 5.040355e+01 3.593556e+01 4.213013e+01 3.193390e+01 2.497008e+01 2.051501e+01 2.155564e+01 2.510994e+01 1.201700e
+00 2.015445e+00 1.348279e+00 1.589908e+00 1.876995e+00 1.810325e+00 1.991898e+00 1.682100e+00 1.658763e+00 1.468302e+00 1.457561e+00 1.262385e+00 1.134145e+00
<GCONST> 9.452953e+01
<TRANSP> 5
```

Fig. 3.18: Trecho da criação do HMM.

```
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 25
 8.893705e-01 -5.965678e-01 -7.401807e-01 -7.936882e-01 -6.253911e-01 -1.767648e-01 -1.675177e-01 -8.361037e-02 -6.576724e-02 4.416337e-02 1.704077e-01 8.305174e-01
-1.988546e-03 3.547536e-03 -9.103909e-03 -1.446508e-02 -4.546112e-03 -8.107636e-05 2.016575e-03 4.418559e-03 4.554304e-03 -2.601711e-03 -5.772511e-04 6.181820e-03
-4.398886e-03
<VARIANCE> 25
 5.025036e+01 5.540667e+01 4.459308e+01 5.833651e+01 5.089742e+01 3.428844e+01 3.919624e+01 3.087931e+01 2.465816e+01 2.108301e+01 2.019370e+01 2.704788e+01 1.222654e
+00 2.146656e+00 1.347326e+00 1.471124e+00 1.851652e+00 1.745831e+00 1.891048e+00 1.724450e+00 1.639278e+00 1.460736e+00 1.441351e+00 1.361490e+00 1.235671e+00
<GCONST> 9.462568e+01
<STATE> 3
<MEAN> 25
 8.775408e-01 -5.720071e-01 -7.838219e-01 -8.624655e-01 -6.456620e-01 -1.714651e-01 -1.531528e-01 -6.502371e-02 -4.778589e-02 3.341082e-02 1.680356e-01 8.562596e-01
-3.969843e-03 5.553602e-03 -1.065093e-02 -1.502818e-02 -2.625284e-03 2.761781e-03 4.676438e-03 3.781133e-03 3.414660e-03 -3.732610e-03 -1.193759e-03 4.919731e-03
-8.434388e-03
<VARIANCE> 25
 5.031723e+01 5.491333e+01 4.434880e+01 5.828866e+01 5.056900e+01 3.393836e+01 3.933645e+01 3.086286e+01 2.455518e+01 2.110126e+01 2.018905e+01 2.703242e+01 1.214749e
+00 2.133789e+00 1.332610e+00 1.458374e+00 1.839817e+00 1.737126e+00 1.880340e+00 1.720837e+00 1.634234e+00 1.463610e+00 1.440102e+00 1.360232e+00 1.229150e+00
<GCONST> 9.453436e+01
<STATE> 4
<MEAN> 25
 8.909574e-01 -5.490478e-01 -8.798320e-01 -1.021265e+00 -6.978059e-01 -1.279356e-01 -1.113586e-01 -6.328277e-02 -4.032155e-02 1.802202e-02 1.580979e-01 9.235603e-01
-6.545860e-03 9.365194e-03 -1.344323e-02 -1.717242e-02 -5.373282e-04 7.088369e-03 1.056690e-02 3.994044e-03 8.247728e-04 -6.481447e-03 -2.745552e-03 3.981989e-03
-1.433057e-02
<VARIANCE> 25
 5.090744e+01 5.477881e+01 4.429930e+01 5.882086e+01 4.972335e+01 3.299162e+01 3.938482e+01 3.048881e+01 2.423007e+01 2.113555e+01 2.016905e+01 2.715984e+01 1.210920e
+00 2.115309e+00 1.312250e+00 1.438617e+00 1.809239e+00 1.718238e+00 1.858652e+00 1.712153e+00 1.621004e+00 1.469582e+00 1.437587e+00 1.354316e+00 1.219299e+00
<GCONST> 9.438399e+01
<TRANSP> 5
```

Fig. 3.19: Trecho da Re-estimação.

A última etapa do treinamento foi a criação do modelo trifone (união de 3 fones). Essa etapa pode ser feita de forma manual, ou seja, juntando 3 fones do seu dicionário, ou utilizando a ferramenta HLEd. As duas formas foram testadas neste projeto, apresentando o mesmo desempenho. A figura 3.20 mostra o modelo trifone.

```
m-i~+n
i~-n+a
n-a+w
a-w
K+o~
K-o~+p
```

Fig. 3.20: Trecho do modelo trifone.

Foram feitas 15 re-estimações obtendo reconhecimento das palavras em torno de 97%.

3.5 Julius

Após realizar a construção do modelo de voz, foi feita a integração do modelo com o decodificador Julius. Isso é possível alterando o arquivo de configuração julius.jconf, “apontando” para a base do modelo de voz criado. Este trabalho utilizou a base do modelo trifone gerado.

Com a integração realizada, foi desenvolvido um sistema para reconhecimento de voz por comando, utilizando a linguagem de programação nativa do UNIX Shell Script. A linguagem foi escolhida por ser nativa do S.O. Linux e por demandar pouco recurso de hardware.

O vocabulário do sistema restringiu-se nas palavras encontrada no modelo acústico proposto. Palavras fora do contexto não foram tratadas.

4 Conclusão

A nova interface via voz está em ampla expansão. Isso ocorre devido a velocidade da interface e pelo aspecto de inclusão que a mesma propõe. Hoje, sabe-se que 24,6 milhões de brasileiros são portadores de deficiência que afetam a visão e movimentação dos membros. Porém, as tecnologias de reconhecimento de voz são feitas para a língua inglesa, inviabilizando o uso pelos brasileiros.

Este projeto explorou o primeiro nível para a construção de um reconhecedor de voz independente de locutor em português brasileiro. Todas as etapas para a modelagem acústica da voz foram abordadas nesse projeto, passando pelas ferramentas necessárias ao tratamento de dados, até a integração com o decodificador Julius. Proporcionando compreensão das etapas realizadas, culminando no desenvolvimento de um sistema de reconhecedor de voz por comandos através de vocabulário pequeno.

Nos testes do sistema, o modelo acústico juntamente com o descritor Julius, apresentaram consistência, funcionalidade para homens e mulheres, e uma taxa média de acerto (com pouco ruído no ambiente) de 97%. Também foram realizados testes na presença de ruído, obtendo taxa média de acerto 43%.

Para trabalhos futuros sugere-se:

- Construção de um reconhecedor de voz contínua independente de locutor, utilizando Wavelet packets e modelada acusticamente sobre a premissa da fonética e fonologia. Como adaptação desse trabalho, sugere-se a aplicação em veículos, biometria por voz e em cadeiras de rodas motorizadas.
- Aplicação intervalar no janelamento da voz.
- Separação das sílabas para o português brasileiro em sistemas de reconhecimento de voz.
- Processamento distribuído para reconhecimento de voz.

Bibliografia

A. ALCAIM, S. C. B. D. S. *Sílabas como unidades fonéticas para o reconhecimento de voz em português*. 2001. SBA Controle & Automação. vol. 12, n. 01.

ALENCAR, V. F. S. de. *Atributos e Domínios de Interpolação Eficiente em Reconhecimento de Voz Distribuido*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2005.

BISOL, L. *Introdução a estudos de fonologia do português brasileiro.* 4. ed. Porto Alegre: EDIPUCRS, 2005.

BISPO, S. C. *Reconhecimento de voz contínua para o português utilizando modelos de Markov escondidos*. Tese (Doutorado) — Programa de Pós-Graduação em Engenharia Elétrica: PUC-Rio, Rio de Janeiro, 1997.

BRESOLIN, A. A. *Reconhecimento de voz através de unidades unidades menores do que a palavra, utilizando Wavelet Packet e SVM em uma nova Estrutura Hierárquica de Decisão*. Tese (Doutorado) — Programa de Pós-Graduação em Engenharia Elétrica: Universidade Federal do Rio Grande do Norte, Natal, 2008.

CARVALHO, J. L. A.; DIAS, D. Técnicas de codificação de voz aplicadas em sistemas móveis celulares. 2000.

CHAN, C. P.; CHING, P. C.; LEE, T. *Noisy speech recognition using de-noised multiresolution analysis acoustic features*. 2001. Journal Acoustical Society of America.

CHING, W.-K.; NG, M. K. *Markov Chains: models, algorithms and applications*. 1. ed. New York: Springer Science + Business Media, 2006.

COOK, S. *Speech Recognition HOWTO*. [S.l.: s.n.], 2002.

CUADROS, C. D. R. *Reconhecimento de voz e de locutor em ambientes ruidosos: comparação das técnicas MFCC e ZCPA*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia da Escola de Engenharia da Universidade Federal Fluminense, Rio de Janeiro, 2007.

DESHMUKH, O.; ESPY-WILSON, C. Y.; JUNEJA, A. *Acoustic Phonetic Speech Parameters for Speaker-Independent Speech Recognition*. 2002. ICASSP-2002. n. 2162. Speech Processing.

GIL, A. C. *Como elaborar projetas de pesquisa*. 4. ed. São Paulo: Atlas, 2002.

HOPCROFT, J. E.; ULLMAN, J. D.; MOTAWANI, R. *Introdução à Teoria de Autômatos, Linguagens e Computação*. 2. ed. São Paulo: Editora Campus, 2008.

HOSOM, J. P. *Automatic Phoneme Alignment Based on Acoustic-Phonetic Modeling*. 2002. International Conference on Spoken Language Processing. ICSLP 02.

HWANG, M.; HUANG, X. *Shared-Distribution Hidden Markov Mod- els for Speech Recognition*. 1991. Speech and Audio Processing.

JIANG, H. E.; MENG, J.; GAO, Y. *Feature extraction using wavelet packets strategy*. 2003. Proceedings 42rd IEEE Conference on Decision and Control.

JUANG, B.; RABINER, L. R. *Hidden Markov Models for Speech Recognition*. 1991. Technometrics.

JÚNIOR, A. H. de S. *Avaliação de redes neurais auto-organizáveis para reconhecimento de voz em sistemas embarcados*. Dissertação (Mestrado) — Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Ceará, 2009.

KIM, K.; YOUN, D. H.; LEE, C. *Evaluation of wavelet filters for speech recognition*. 2000. IEEE International Conference on Systems, Man, and Cybernetics.

LADEFOGED, P. *A course in phonetics*. 4. ed. [S.l.]: University of California, Los Angeles, 2001.

LATSCH, V. L. *Construção de Banco de Unidades para Síntese da Fala por Concatenação no Domínio Temporal*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, COPPE , Rio de Janeiro, 2005.

MALBOS, F.; BAUDRY, M.; MONTRESOR, S. *Detection of stop consonants with the wavelet transform*. 1994. Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis.

MARCHESI, B.; LIPPMANN, L. J.; NOHAMA, P. *Voice recognition method applied to Brazilian vowels*. 1996. Proceedings of the 18th Annual International Conference of the IEEE. Engineering in Medicine and Biology Society.

MARTINS, J. A. *Avaliação de diferentes técnicas para reconhecimento da fala*. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 1997.

MORAIS, E. S. *Reconhecimento automático da fala contínua empregando modelos híbridos ANN + HMM*. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação da UNICAMP, Campinas, 1997.

MUSSALIM, F.; BENTES, C. A. *Introdução à linguística: domínios e fronteiras*. 2. ed. [S.l.]: São Paulo, 2001.

OPPENHEIM, A. V.; WILLSKY, A. S.; NAWAB, S. H. *Signals & systems*. 2. ed. USA: Upper Saddle River, NJ, 1996.

RABINER, L. R.; JUANG, B. *Fundamentals of Speech Recognition*. 1. ed. USA: New Jersey, Prentice Hall, 1993.

SAADEQ, R. M.; ALI, A. K. *A novel model characteristics for noise-robust Automatic Speech Recognition based on HMM*. 2010. International Conference of the IEEE. Wireless Communication, Network and Information Security (WCNIS).

SANTOS, D. A. O. *Reconhecimento de voz em presença de ruído*. Dissertação (Mestrado) — ontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001.

SARIKAYA, R.; GOWDY, J. N. *Subband based classification of speech under stress*. 1998. Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 98.

SIMOES, F. O. *Implementação de um Sistema de Conversão Texto-Fala para o Português do Brasil*. Dissertação (Mestrado) — UNICAMP, Campinas, 1999.

SOLEWICZ, J. A.; MORAES, J. A.; ALCAIM, A. *Text-to-speech system for brazilian portuguese using a reduced set of synthesis units*. 1994. International Symposium on Speech, Image Processing and Neural Networks.

SOTOMAYOR, C. A. M. *Realce de Voz Aplicado à Verificação Automática de Locutor*. Dissertação (Mestrado) — Instituto Militar de Engenharia, 2003.

TAN, B. T. et al. *The use of wavelet transforms in phoneme recognition*. 1996. Fourth International Conference, on Spoken Language, ICSLP 96.

TOLBA, H. *Comparative Experiments to Evaluate the Use of Syllables for the Improvement of Automatic Recognition of Dysarthric Speech*. 2009. IEEE 16th International Conference on Systems, Signals and Image Processing.

VALIATI, J. F. *Reconhecimento de Voz para Comandos de Direcionamento por meio de Redes Neurais*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, 2000.

YNOGUTI, C. A. *Reconhecimento da fala contínua usando o modelo oculto de Markov*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, 1999.

YOUNG, S. *The HTK hidden Markov model toolkit: Design and philosophy*. [S.l.], 1994. Disponível em: <http://htk.eng.cam.ac.uk>.

YOUNG, S. *A Review of Large-Vocabulary Continuous-Speech Recognition*. 1996. IEEE Signal Processing Magazine.

ZHU, Q.; ALWAN, A. *On the use of variable frame rate analysis in speech recognition*. 2000. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 00.

APÊNDICE A

CÓDIGO FONTE DA APLICAÇÃO(RECONHECIMENTO_VOZ)

```
#!/bin/bash

h=1

interface=$(zenity –title “ ****Bem vindo***** ” –info –text “Reconhecedor de voz 100% PT_BR”)

desejo=$(zenity –title=“Computador pronto!” –list –text “O que você deseja?” –radiolist –column “Escolha” –column “Desejo” TRUE “Ligar Reconhecedor” FALSE “Ajuda”)

if [[ $desejo = “Ajuda” ]]

then

gedit ajuda

exit 1

fi

while true ; do

cat reconhece | grep sentence1 | tr ’[A-Z]’ ’[a-z]’ | sed ’s/sentence1: <s>//g’ | cut -d “<” -f1 | cut -d “ ” -f3 > temporario

b=$(cat -n temporario | grep $h | sed ’s/’$h’//g’)

if [[ $b != “ ” ]]

then

if [ $b = “desligar” ]

then

rm -rf reconhece

rm -rf temporario

killall julius
```

```
break
exit 1
fi
if [ $b = “editor” ]
then
gedit
fi
if [ $b = “terminal” ]
then
gnome-terminal
fi
if [ $b = “navegador” ]
then
firefox
fi
if [ $b = “musica” ]
then
vlc “$HOME/by”
fi
if [ $b = “quente” ]
then
stellarium
fi
let h++
fi
sleep 3
done
```

CÓDIGO FONTE DA APLICAÇÃO(JULIUS)

```
julius -input mic -C ../julius.jconf > reconhece
```

CÓDIGO FONTE DA APLICAÇÃO(AJUDA)

*****Bem Vindos ao Reconhecedor de palavras Isoladas******

O Programa foi treinado para reconhecer os comandos de acordo a regra:

-> <CHAMADA> <NOME>

-> <DIA> <TEMPO>

Onde:

<CHAMADA> é representada por:

* computa

* computador

<NOME> é representada por:

* desligar -> Encerrar o Programa

* navegador -> Abre o Firefox

* terminal -> Abre o gnome-terminal

* editor -> Abre o Gedit

<DIA> é representada por:

* segunda

* terça

* quarta

* quinta

* sexta

* sábado

* domingo

<TEMPO> é representada por:

* quente -> Abre o Programa Stelarium

CÓDIGO FONTE DA APLICAÇÃO(MAIN)

```
#!/bin/bash

./reconhecimento_voz &

./julius &
```