UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# Desenvolvimento de Recursos para a Construção de um Sistema Texto-Fala para o Português Brasileiro

**Igor Costa do Couto**

UFPA/ITEC/PPGEE

Campus Universitário do Guamá

Belém-Pará-Brasil

2010

UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# Desenvolvimento de Recursos para a Construção de um Sistema Texto-Fala para o Português Brasileiro

**Autor: Igor Costa do Couto**

**Orientador: Aldebaro Barreto da Rocha Klautau Júnior**

**Dissertação** submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará para obtenção do Grau de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Telecomunicações**.

UFPA/ITEC/PPGEE

Campus Universitário do Guamá

Belém, PA

2010

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# Desenvolvimento de Recursos para a Construção de um Sistema Texto-Fala para o Português Brasileiro

AUTOR: IGOR COSTA DO COUTO

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

---

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior
(Orientador - UFPA)

---

Prof. Dr. Ronaldo de Freitas Zampolo
(Membro - UFPA)

---

Prof. Dra. Valquíria Gusmão Macedo
(Membro - UFPA)

---

Prof. Dr. Yomara Pinheiro Pires
(Membro - UFPA)

UFPA/ITEC/PPGEE

# Agradecimentos

Agradeço a minha família, em especial a minha mãe, Irene, que não mede esforços e nem economiza disposição para me abrir caminhos à vitória. Também a minha irmã, Giuliane, que me deu apoio nos momentos difíceis e teve paciência ao longo desta etapa. Aos meus avós, tios e primos, pois são pessoas essenciais para mim. Ao meu pai, que mesmo ausente, deu-me oportunidades. E ao meu afilhado, Pedro Henrique, apesar da pouca idade.

A todos os amigos e colegas de trabalho do Laboratório de Processamento de Sinais (LaPS) do grupo FalaBrasil: Patrick, Fabíola, Nelson, Carol, Renan, Pedro, Denise, Vincent e Anderson não só por se mostrarem sempre solícitos em ajudar, mas também pelo companheirismo. Ao meu orientador, Prof. Dr. Aldebaro Klautau Jr., tanto pela orientação durante minha breve passagem pela academia como pela boa vontade, amizade e bom trabalho desenvolvido. Aos amigos Claudomir, Muller, Yomara, Jefferson, Adalbery, Fernanda e Kelly pela descontração e momentos agradáveis que me proporcionaram. Aos demais amigos do LaPS, agradeço pela convivência.

Ao Prof. Dr. Ranniery Maia, que mesmo distante (Reino Unido), contribuiu com o trabalho de forma positiva.

A CAPES e Fapespa, por terem me dado incentivo financeiro ao longo destes dois anos.

*Dedico este trabalho aos meus pais e*

*à minha irmã*

*que torceram pelo meu sucesso...*

# Resumo

Sistema Texto-Fala (TTS) é atualmente uma tecnologia madura que é utilizada em muitas aplicações. Alguns módulos de um sistema TTS são dependentes do idioma e, enquanto existem muitos recursos disponíveis para a língua inglesa, os recursos para alguns idiomas ainda são limitados. Este trabalho descreve o desenvolvimento de um sistema TTS completo para Português Brasileiro (PB), o qual também apresenta os recursos já disponíveis. O sistema usa a plataforma MARY e o processo de síntese da voz é baseado em cadeias escondidas de Markov (HMM). Algumas das contribuições deste trabalho consistem na implementação de silabação, determinação da sílaba tônica e conversão grafema-fonema (G2P). O trabalho também descreve as etapas para a organização dos recursos desenvolvidos e a criação de uma voz em PB junto ao MARY. Estes recursos estão disponíveis e facilitam a pesquisa na normalização de texto e síntese baseada em HMM para o PB.

**PALAVRAS-CHAVES:** sistemas texto-fala, TTS, síntese de voz, cadeias escondidas de Markov, normalização de texto, português brasileiro.

# Abstract

Text-to-speech (TTS) is currently a mature technology that is used in many applications. Some modules of a TTS depend on the language and, while there are many public resources for English, the resources for some underrepresented languages are still limited. This work describes the development of a complete TTS system for Brazilian Portuguese (BP) which expands the already available resources. The system uses the MARY framework and is based on the hidden Markov model (HMM) speech synthesis approach. Some of the contributions of this work consist in implementing syllabification, determination of stressed syllable and grapheme-tophoneme (G2P) conversion. This work also describes the steps for organizing the developed resources and implementing a BP voice within the MARY. These resources are made available and facilitate the research in text normalization and HMM-based synthesis for BP.

**KEYWORDS:** text-to-speech systems, TTS, speech synthesis, hidden Markov models, text normalization, Brazilian Portuguese.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Glossário

| | | |
|---|---|---|
| TTS | - | *Text-to-Speech System* |
| HMM | - | *Hidden Markov Models* |
| PB | - | português brasileiro |
| MARY | - | *Modular Architecture for Research on speech sYnthesis* |
| G2P | - | *Grapheme-to-Phoneme* |
| IPA | - | *International Phonetic Alphabet* |
| F0 | - | frequência fundamental |
| PLN | - | Processamento de Linguagem Natural |
| PDS | - | Processamento Digital de Sinais |
| SAMPA | - | *Speech Assessment Methods Phonetic Alphabet* |
| MFCC | - | *Mel-frequency cepstral coefficients* |
| MLSA | - | *Mel Log Spectrum Approximation* |
| WER | - | *Word Error Rate* |
| WAR | - | *Word Accuracy Rate* |

# Capítulo 1

# Introdução

Desde a pré-história humana até os dias atuais, a interação através da fala tem sido o modo dominante pelo qual o homem tem trocado informação. A fala hoje pode ser facilmente difundida, através de diversas maneiras, como rádio, telefone, filmes e, mais recentemente, Internet. E outras formas com certeza surgirão. Segundo [1], esta tendência reflete a supremacia que a comunicação falada tem nas relações humanas.

Durante anos, a fala serviu somente para a comunicação homem-homem, deixando os computadores - enraizados há algumas décadas na sociedade - fora desta relação. A interação entre homem-máquina era unicamente visual, através das interfaces gráficas do usuário (*GUI's*), representadas por janelas, menus, objetos, ponteiros e ícones nos monitores. Com as constantes evoluções de tecnologias que gerassem voz sintetizada no computador, as máquinas também passaram a se beneficiar da situação. Mais do que tornar o computador um falante natural, o homem também busca integrar cada vez mais aquelas pessoas que não podem trabalhar à moda antiga (monitor e teclado) com o computador.

O nosso mundo mudou radicalmente nas últimas décadas, devido à explosão de novas tecnologias e ao aparecimento da Internet, o que criou novos paradigmas de acesso à informação e ao conhecimento. A informação, agora, está disponível em uma rede gigantesca e de forma abundante, dificultando a navegação. Por conseguinte, encontrar meios para utilizá-la proveitosamente acabou por se tornar um novo desafio. É nesse intervalo entre sociedade e informação (homem-máquina), que as tecnologias de fala se encaixam perfeitamente como uma ferramenta para potencializar os benefícios dessa revolução tecnológica.

Grandes companhias, como a revolucionária Google, proprietária do navegador web Chrome, já investem fortemente em tecnologias de voz para tentar consolidar seus produtos

como padrão na indústria e na interação do usuário e com o computador através da fala[1]. Outro exemplo é a Microsoft, que já integra interfaces de fala em muitos dos seus aplicativos. No endereço *web* da empresa, é visto uma frase que explica bem o cenário: "Nossa visão: Interfaces de fala que fazem programas e serviços fáceis e mais naturais para o uso."[2]. Estas empresas sabem que se manter na vanguarda tecnológica sempre é um diferencial.

Ainda que a indústria esteja voltada para a pesquisa destas tecnologias, é evidente que nem todas as línguas recebem a mesma atenção. Trabalhos mais extensos em Português ainda são escassos quando comparados com o portfólio de línguas de maior presença atual na academia, como o inglês. Muito se deve ao status econômico desta última. Apesar de ser apresentada como um problema, é essa falta de atenção para o Português Brasileiro (PB) que se tornou um dos motivos para iniciar a investigação presente nesta dissertação.

## 1.1 Contexto

Esta dissertação aborda a questão de como fazer as máquinas lerem um texto e depois falarem com a naturalidade típica dos seres humanos. Os campos de estudo que cuidam das questões envolvidas neste desafio são conhecidos como (1) **síntese de voz**, que é a geração de voz sintética, e (2) **sistemas texto-fala** [2, 3], que é o processo de converter texto escrito em fala. Estas coexistem com outras tecnologias de voz como **reconhecimento de fala**, que visa transformar voz em texto, e a aplicação em **tradução automática**, que faz a conversão de escrita ou fala entre diferentes línguas. A Figura 1.1 contextualiza a área de síntese de voz dentro do processamento de voz, junto com o reconhecimento de voz.

O desafio pode ser particionado em três problemas: o processo de leitura, o processo da fala e as questões envolvidas em tornar o computador uma entidade capaz de realizar os dois procedimentos anteriores. O interessante é que alguns estudiosos em psicologia chegam a afirmar que os comportamentos humanos (ler e falar) não poderiam ser reduzidos em um sistema de informação [4], tal qual um computador. Tais sistemas deveriam ser vistos apenas como dispositivos, aparelhos ou tecnologias que oferecem suporte à comunicação humana e nada mais.

Se é possível, ou não, dotar uma máquina de comportamento humano pouco importa aqui, o fato é que, atualmente, pode-se considerar corriqueiro ouvir vozes sintetizadas. Com o avanço nas pesquisas, é provável que mais e mais destas tecnologias estejam na sociedade

[1] http://www.infoworld.com/d/developer-world/google-building-speech-capabilities-browsers-071?source=footer

[2] http://www.microsoft.com/speech/

sem que se tenha a capacidade de saber o que é homem ou computador. Os sistemas de atendimento automático de grandes corporações já adotam reconhecedores e sintetizadores de fala e podem servir de bom exemplo.

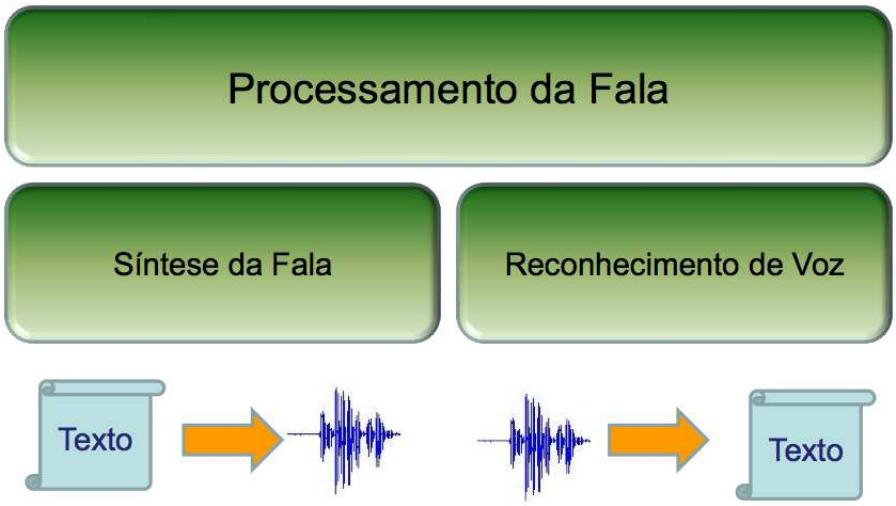


Figura 1.1: Contextualização das áreas de pesquisa relacionadas com o trabalho.

Neste contexto, este trabalho se dispõe, junto com o grupo de pesquisa em processamento de voz *FalaBrasil* [5], a desenvolver e disponibilizar recursos para o PB, de forma a criar um completo sistema de referência e permitir que outros grupos de pesquisa e desenvolvedores de *software* se beneficiem dos recursos disponibilizados. Criado em 2009 pelo Laboratório de Processamento de Sinais (LaPS) da Universidade Federal do Pará (UFPA), o *FalaBrasil* focou inicialmente em ferramentas destinadas ao reconhecimento de voz, sendo que este trabalho representa o primeiro passo em direção a aplicações voltadas para a síntese de voz.

## 1.2 Antecedentes e trabalhos relacionados

Na academia, o primeiro sistema texto-fala completo para o PB apareceu no final da década de 90, na Universidade de Campinas (UNICAMP) e na Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ) com a síntese baseada em formantes [6] e a concatenativa (difone) [7–10]. Atualmente, testes de escuta informais indicam que as mais maduras plataformas estão sendo desenvolvidos na Universidade Federal de Santa Catarina (UFSC) [11]. Em [12], um motor de síntese (*back end*) baseado em HMM é apresentado, sendo que os autores também disponibilizaram o *corpus* e os códigos para rotinas de treino em [13].

Na indústria, existem companhias que oferecem vozes sintetizadas para o PB com o uso de *engines* específicas. Entre eles podemos citar Raquel da Nuance [14], Fernanda, Gabriela e

Felipe da Loquendo [15] e Marcia, Paola e Carlos da Acapela [16]. A Microsoft também tem suporte para o PB em dispositivos móveis e computadores pessoais [17].

Com exceção de [12], os sistemas mencionados não estão disponíveis, por exemplo, para propósitos de pesquisa. Focando na metodologia de código aberto, a plataforma para o PB do projeto MBROLA [18] e o sistema DOSVOX [19] merecem ser mencionados. O MBROLA é um projeto que tem o objetivo de se tornar um sintetizador de fala multi-língua. Já o DOXVOX corresponde a um sistema operacional livre para os deficientes visuais e inclui seu próprio sintetizador de voz, além de oferecer a possibilidade de utilizar outros mecanismos. Outro recurso relevante é o conjunto de ferramentas do CSLU [20], o qual possui suporte à síntese baseada em difone pela voz AGA.

## 1.3 Objetivos

Este trabalho tem como principal objetivo o desenvolvimento de recursos para a construção de um sistema TTS para o PB, assim como a sua integração para a criação de uma plataforma completa no futuro. Também relacionado ao trabalho, podem ser listados os seguintes objetivos específicos:

- Desenvolver o estudo de tecnologias TTS para o PB.
- Fortalecer a pesquisa nacional, tanto na indústria quanto na academia.
- Fornecer recursos específicos para o PB.
- Gerar módulos que futuramente possam servir de suporte para implementar um sistema TTS completo e livre.
- Aumentar o portfólio dos recursos disponibilizados pelo projeto *FalaBrasil*.
- Ajudar a consolidar a UFPA como uma referência no estudo de tecnologias de linguagem (reconhecimento e síntese) para o PB.

## 1.4 Contribuições

Seguindo a premissa do projeto *FalaBrasil*, os códigos e módulos implementados foram disponibilizados gratuitamente para a comunidade [5]. Ainda que o sistema não esteja fechado (completo e imutável), todas as partes necessárias para a composição são acessíveis e serão documentadas aqui. Sintetizando algumas das contribuições do trabalho, tem-se:

- Conversor de símbolos e caracteres especiais.
- Expansor de abreviaturas.
- Expansor de siglas.
- Conversor de numerais.
  - Conversor de números árabes cardinais.
  - Conversor de números árabes ordinais.
  - Conversor de números romanos.
- Conversor Grafema-fonema.
- Divisor silábico
- Marcador de sílaba tônica.
- Alfabeto fonético com as características distintivas.

Além dos algoritmos e respectivas implementações citadas acima, também gerou-se um artigo no Simpósio Internacional de Telecomunicações (International Telecommunications Symposium 2010 - ITS2010), ocorrido em Manaus, no estado do Amazonas, de 6 a 9 de setembro. O documento, além de descrever uma parte desta dissertação, é um guia rápido em que outros pesquisadores podem se basear para construir seus próprios sistemas TTS.

- An Open Source HMM-based Text-to-Speech System for Brazilian Portuguese. Couto, I. ; N. Nelson; V. Tadaiesky; M. Ranniery; Klautau, A. International Telecommunications Symposium (ITS2010), 2010.

O trabalho ainda gerou uma voz sintetizada, baseada na base de dados disponibilizada pelo pesquisador Dr. Ranniery Maia [12]. A voz foi construída em cima da plataforma ***M****odular* ***A****rchitecture for* ***R****esearch on speech s* ***Y****nthesis* (MARY) [21], que nos últimos anos tem se tornado uma famosa ferramenta de ensino e, principalmente, pesquisa na área de síntese de voz.

## 1.5 Síntese dos capítulos

O restante desta dissertação encontra-se organizada em 4 partes::

- Capítulo 2: SÍNTESE DE VOZ. Em qualquer ciência, conceitos e fundamentos teóricos são requisitos necessários para se aprofundar em algum tema ou assunto. Neste capítulo, o leitor adquire os conhecimentos essenciais para a compreensão do problema específico apresentado e da solução proposta.

- Capítulo 3: ALGORITMOS PARA A ANÁLISE DE TEXTO. Das etapas de um sistema TTS, sem dúvida alguma, os módulos iniciais são os mais sensíveis à mudanças de idioma. Tais componentes geralmente implementam algoritmos e regras que são dependentes a língua em questão. Para o PB, esta é uma das etapas que mais merecem atenção, já que é raramente explorada e pouco documentada pela academia e indústria.

- Capítulo 4: CONSTRUÇÃO DO SISTEMA TTS. Desenvolver um programa que converta texto em áudio sem recursos iniciais é considerada uma tarefa trabalhosa. Acrescenta-se ainda o agravante de que língua para a qual o sistema será desenvolvido (PB) é carente de recursos e documentação que guiem os interessados no assunto, e não tendo o mesmo apelo comercial de idiomas culturalmente mais explorados, como o inglês. Nesta parte da pesquisa, será apresentada a solução (não definitiva e mutável) da construção do sistema TTS, unindo o conhecimento adquirido e as ferramentas construídas nos capítulos anteriores, junto à plataforma MARY.

- Capítulo 5: CONSIDERAÇÕES FINAIS. Uma síntese do trabalho, além de uma avaliação do que foi feito e dos pontos que ainda podem receber melhorias e se beneficiar de novas abordagens, visando construir um sistema mais robusto.

# Capítulo 2

# Síntese de voz

No presente capítulo serão apresentados os principais conceitos e aspectos envolvendo síntese de voz, focando especificamente em sistemas TTS. Nestes *softwares*, da escrita até a formação da fala, um processo muito complexo é realizado ao longo de diversas etapas. E como o problema não é recente, várias abordagens foram criadas para a se chegar a uma solução, inclusive misturando várias técnicas [22–24]. Engenharia, linguística, informática e matemática formam uma lista não-exaustiva de áreas da ciência em que tais sistemas se embasam para tentar fazer com que as máquinas reproduzam voz mais próxima do natural.

## 2.1 Conceitos gerais

Como todas as áreas de estudo, a síntese de voz também é composta com um conjunto termos, ideias e conceitos. Devido a isto, é uma boa prática identificar e explicar pelo menos alguns itens deste conjunto. A intenção desta seção não é se aprofundar e, sim, passar alguns aspectos gerais que são de interesse do leitor e que são necessários para a compreensão do que foi desenvolvido neste trabalho. Começa-se dos conceitos básicos sobre a fala e conclui-se com as técnicas utilizadas em síntese de voz.

### 2.1.1 Linguagem, língua e fala

Segundo [25], a **linguagem** é a habilidade de expressar os pensamentos de um ser através de sinais, que podem ser gráficos (escrita), gestuais (linguagem dos surdos-mudos), acústicos (voz) e até mesmo musicais. A **língua** é um tipo de linguagem, sendo a única modalidade de linguagem baseada em palavras. É o conjunto de sinais que permitem a uma

pessoa compreender e fazer-se compreender [26]. Por exemplo, o alemão e o português são línguas diferentes. Por último vem a **fala**, que é a realização concreta (mecanismo físico) da língua, feita por um indivíduo num determinado momento com o intuito de expressar ideias. É um ato individual que cada membro pode efetuar com o uso da linguagem. A fala é um dos principais componentes para diferenciar os seres humanos. Em [27], o autor afirma que o principal objetivo da fala humana é a troca de ideias.

Apesar de intimamente relacionadas, voz e fala possuem diferenças [26]. **Voz** é o conjunto de sons emitidos pelo aparelho fonador que pode variar em altura, frequência, timbre, etc. Já a fala seria o uso desta habilidade para emitir sons em um padrão (língua) para expressar algum pensamento pessoal. Segundo Saussure [26], a fala seria um mecanismo psico-físico que permite ao ser humano exteriorizar ideias. Sendo assim, a voz seria somente o aspecto físico em questão. Neste trabalho, não serão levadas em conta tais diferenças entre voz e fala, mas que fique claro para o leitor que estas existem.

### 2.1.2 Níveis da fala

A informação extraída da fala pode ser analisada em vários planos, gerando abstrações que são importantes e que ajudam a entender o funcionamento dos sistemas TTS. A literatura geralmente distingue vários níveis de voz não mutualmente exclusivos. É válido lembrar que definir a voz em níveis pode nos levar a crer que esta é facilmente dividida, porém, as fronteiras aqui demarcadas nem sempre são tão claras. Segundo Dutoit [2], os níveis da fala são: acústico, fonético e fonológico, morfológico, sintático e semântico. A Figura 2.1 ilustra a organização destes níveis.


Figura 2.1: Níveis da fala. Adaptado de [28].

#### 2.1.2.1 Nivel acústico

A voz (ou fala) é, fisicamente, uma variação na pressão do ar causada e emitida por um sistema articulatório. Embora os sons da fala tenham muitas propriedades características, melhor especificadas no espectro (frequência) do que em formas de onda (tempo), existe uma grande variabilidade de relações entre sinais acústicos e unidades linguísticas representativas - estas últimas discretas, como palavras, sentenças, frases, etc. Sinais acústicos são contínuos, sendo difícil mapear as fronteiras de palavras (ou qualquer outra unidade) em intervalos na fala.

O estudo da fala comumente passa pela transformação do sinal acústico em sinais elétricos através de um microfone. Nos sistemas mais atuais, o sinal é ainda digitalizado, ou seja, é filtrado, amostrado e quantizado. A partir deste armazenamento é possível realizar várias operações de processamento de sinais para descobrir parâmetros relacionados à este nível: **frequência fundamental**, **intensidade** e **distribuição espectral de energia**. A quantidade perceptual relacionada a cada um desses aspectos acústicos é respectivamente chamada de: ***pitch***, **altura** e **timbre** [2].

O ***pitch*** é a percepção da? frequência fundamental dos sons. A faixa de som que o ser humano escuta varia de 20 a 20000 Hz, mas a maior sensitividade se concentram nas frequências entre 200 e 2000 Hz.

A **altura** é o modo como a amplitude (ou intensidade) é percebida. Uma mudança em amplitude não necessariamente é sentida proporcionalmente na altura, pois esta é influenciada tanto pela frequência como pelo timbre do som.

O **timbre**, ou qualidade do tom, é a nossa percepção relacionada ao fenômeno físico de desdobramentos parciais do espectro do som, chamado de envelope espectral. É o que nos permite distingir dois instrumentos diferentes tocando a mesma nota na mesma amplitude, pois eles não irradiam o espectro igualmente em todas as direções. Os primeiros estudos sobre o timbre datam do Século XIX no livro de Helmholtz chamado *Sensations of Tone*. Estudos do timbre são fundamentais para os músicos.

Existem informações muito valiosas no nível acústico. Por exemplo, em sintetizadores baseados em HMM [29], os parâmetros utilizados pelo modelo não são diretamente os valores do sinal no tempo e sim a frequência fundamental e coeficientes cepstrais da escala mel [30], que são parâmetros obtidos no nível acústico.

#### 2.1.2.2 Nivel fonético e fonológico

O estudo de como os sons humanos são produzidos e utilizados na língua é uma disciplina científica bem estabelecida, e com um robusto conhecimento teórico. Segundo [27], este campo é dividido em dois ramos: a geração e classificação dos sons da fala é abrangida pelo tema da **fonética** [31], já as funções destes sons interessam à **fonologia** [31]. Estes dois campos não necessitam ser estudados em detalhes pelos que tem interesse nas tecnologias de fala, porém alguns aspectos devem ser apreciados em linhas gerais. Como será visto em capítulos posteriores, para entender como funcionam as técnicas de síntese de voz é necessário ter algum conhecimento sobre termos básicos da fonética e fonologia.

Ainda pode-se definir as duas áreas do nível fônico de outras maneiras. A fonética, em linhas gerais, é a responsável pelo estudo de quaisquer sons de uma determinada língua, o que significa admitir que ela se ocupa das diversas variantes de um mesmo som. Assim, por exemplo, pode-se dizer que, ao se fazer a fonética da Língua Portuguesa, preocupa-se em estudar sons de uma determinada letra, levando-se em consideração as suas variantes. A fonologia, por sua vez, não se interessa por todos os sons que existem numa determinada língua. Para esta última, interessarão apenas os sons que têm função comunicativa. Jakobson relaciona função comunicativa com os aspectos sonoros que se utilizam para veicular uma determinada mensagem [32].

A fonética também se preocupa com as características físicas, articulatórias e perceptivas da produção e percepção dos sons da fala, e fornece métodos para a sua descrição e classificação. Para que seja possível a emissão da voz, o aparelho fonador (ou aparelho articulatório), visto na Figura 2.2, transforma o ar que vem dos pulmões em som articulado. Este processo tem início quando esta corrente de ar percorre os brônquios, penetra na traqueia e atinge a laringe, onde poderá encontrar o primeiro obstáculo. Após atravessar a glote, que está localizada na altura do chamado pomo-de-adão, encontrará as pregas vocais, que nada mais são do que duas pregas musculares, que poderão estar abertas ou fechadas. Estando abertas, esta corrente não possuirá barreiras neste trecho do percurso. Entretanto, estando fechadas, o ar forçará a passagem. Tal esforço causará vibração nas pregas e repercutirá em som.

Este som manterá o percurso e encontrará o segundo obstáculo. Ao adentrar a faringe, encontrará duas vias de acesso ao meio externo: cavidades bucal e nasal. Quem determinará o destino deste som será a úvula. De acordo com a posição que a úvula adotar, o som irá atravessar somente o canal bucal ou ambos os canais. Assim, estando a úvula levantada, isto é, unida à parede posterior da faringe, o canal nasal será obstruído e o som terá por caminho somente o canal bucal. Estando abaixada, a corrente de ar irá se dividir e ressoará por ambos

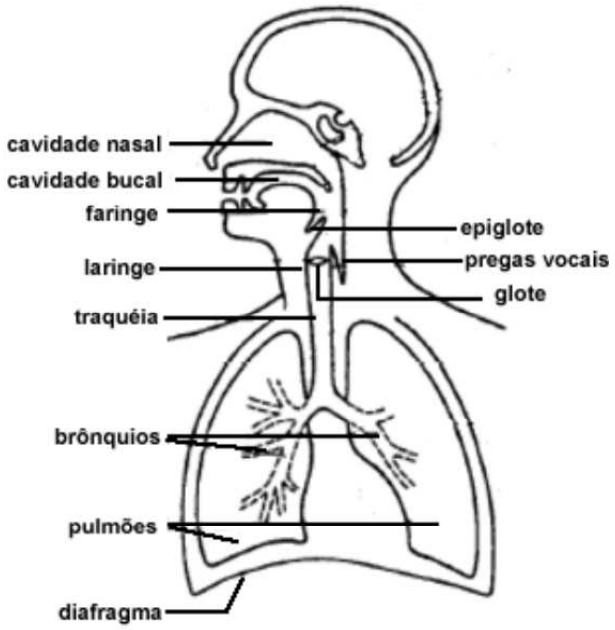


Figura 2.2: Aparelho fonador humano [30].

os canais. Os órgãos encontrados nestes canais serão responsáveis por dar forma ao som, isto é, transformá-lo em voz humana.

De todo este processo, nascerá um outra unidade muito conhecida no estudo da fala: os fonemas. Estes serão melhor explorados na Seção 2.1.3.

#### 2.1.2.3 Nível Morfológico

Morfologia é a parte da linguística que descreve a forma das palavras como uma função de um conjunto reduzido de unidades significativas, chamados **morfemas** e subsequentemente separados em radicais e afixos (sufixos e prefixos). A peculiaridade da morfologia é estudar as palavras olhando para elas isoladamente e não dentro da sua participação na frase ou período. Por exemplo, $per + correr = percorrer$.

#### 2.1.2.4 Nível Sintático

A sintaxe é a parte da linguística que contém as regras relativas à combinação das palavras em unidades maiores (como as orações) e as relações existentes entre as palavras dentro dessas unidades. No português, por exemplo, a ordem mais comum de estruturação das sentenças é $sujeito+verbo+objeto$, como em "*Os brasileiros gostam de futebol*". O sujeito é Os brasileiros, o verbo é gostam e futebol é o objeto.

#### 2.1.2.5 Nível Semântico

Enquanto o nível sintático restringe as formas de estruturação das sentenças, esta ainda não constitui um critério exaustivo de aceitabilidade, pois muitas sentenças podem respeitar as regras gramaticais mas não constituir nenhum significado para o ser humano. Por exemplo, a mesma estruturação $sujeito + verbo + objeto$ pode ser utilizada para compor "*A vida choveu cachorros*", uma sentença totalmente incompreensível. Para isso, a semântica é a parte da linguística que estuda o significado e a aplicação das palavras.

### 2.1.3 Fone/Fonemas

A unidade básica da fonética é o **fone**. O Dicionário de Termos Linguísticos [33], define fone: a menor unidade discreta, perceptível num contínuo sonoro, que constitui a especificação fonética do som de fala. É portanto uma unidade concreta, a realização física de unidades de outro nível, os fonemas. Para **fonema**, o dicionário especifica: a menor unidade do sistema fonológico de uma língua, sendo associado a um contexto. As diferenças entre fone e fonema estão no mesmo nível de distinção entre fonética e fonologia. O primeiro mais associado aos aspectos físicos e o segundo se aproximando das funções que os sons exercem na língua.

Os fonemas são classificados como vogais, cuja distinção baseia-se no processo de formação do som quanto à existência ou não de obstáculos à passagem do ar, consoantes e semivogais. As vogais são fonemas produzidos quando o ar expelido do pulmão faz vibrar os cordões vocais e não sofre obstáculos até sua saída pela boca. Esses fonemas constituem a base da sílaba, pois não dependem de outros fonemas para serem produzidos [34].

Apesar de não representarem obstáculos, alguns componentes do aparelho fonador alteram algumas características dos fonemas produzidos. A influência do véu palatino produz as vogais orais (i, e, é, a, ó, o, u), quando o ar passa somente pela cavidade bucal como nas palavras pá ou foca, e as vogais nasais, quando o ar, além de passar pela cavidade bucal, passa também pela nasal produzindo os fonemas vocálicos nasais presentes em palavras como linda, rã e onda. De acordo com [34], as vogais podem ser classificadas quanto:

1. A Zona de Articulação
   (a) *Anteriores ou Palatais:* Articuladas com a língua elevada em direção ao palato duro, próximo ao dentes, ex: p***é***.
   (b) *Médias:* Articulada com a língua abaixada, quase em repouso, ex: p***a***sta.
   (c) *Posteriores ou Velares:* Articuladas quando a língua se dirige ao palato mole, ex: res***u***mo.

2. A Intensidade

   (a) *Tônicas:* Pronunciada com maior intensidade, ex: c***a***sa.

   (b) *Átonas:* Pronunciada com menor intensidade, ex: cas***a***.

3. Ao Timbre

   (a) *Abertas:* Maior abertura do tubo vocal, ex: p***é***.

   (b) *Fechadas:* Menor abertura do tubo vocal, ex: v***ê***.

   (c) *Reduzidas:* Vogais átonas reduzidas no timbre, ex: car***a***.

4. Ao Papel das Cavidades Bucal e Nasal

   (a) *Orais:* Ressonância apenas na boca, ex: t***u***do.

   (b) *Nasais:* Ressonância em parte da cavidade nasal, ex: m***u***ndo.

Consoantes são os sons produzidos sob a influência de obstáculos à passagem do ar, tais como boca, língua, dentes. Devem ser acompanhadas por pelo menos uma vogal, por não terem pronúncia própria como as vogais. De acordo com [34], as consoantes podem ser classificadas quanto ao:

1. Modo de Articulação

   (a) *Oclusivas:* Corrente expiratória encontra um obstáculo total, que impede a saída do ar, explodindo subitamente, ex: ***p***ai.

   (b) *Constritivas:*

      i. *Fricativas:* Corrente expiratória passa por uma estreita fenda, o que produz um ruído comparável a uma fricção, ex: ***f***aca.

      ii. *Laterais:* Ponta ou dorso da língua se apoia no palato, saindo a corrente de ar pelas fendas laterais da boca, ex: ***l***ua.

      iii. *Vibrantes:* Ponta da língua mantém contato intermitente com os alvéolos, o que acarreta um movimento vibratório rápido, abrindo e fechando a passagem à corrente expiratória, ex: ***c***aro.

2. Ponto de Articulação

   (a) *Bilabiais:* Encontro dos lábios, ex: ***p***ai.

   (b) *Labiodentais:* Encontro do lábio inferior com os dentes superiores, ex: ***f***aca.

   (c) *Linguodentais:* Encontro da ponta da língua com os incisivos superiores, ex: ***t***atu.

   (d) *Alveolares:* Encontro da ponta da língua com os alvéolos dos dentes superiores, ex: ***l***ua.

   (e) *Palatais:* Encontro do dorso da língua com o palato duro, ex: ***x***is.

   (f) *Velares:* Encontro da parte posterior da língua com o palato mole ou véu palatino, ex: ***c***asa.

3. Papel das Pregas Vocais

   (a) *Surdas:* Produzidas sem vibração das pregas vocais, ex: ***f***aca.

   (b) *Sonoras:* Produzidas por vibração das pregas vocais, ex: ***b***ola.

4. Papel das Cavidades Bucal e Nasal

   (a) *Orais:* Ressonância apenas na boca, ex: ***b***ola.

   (b) *Nasais:* Ressonância em parte da cavidade nasal, ex: ***m***ãe.

Há casos onde a diferenciação de uma consoante com a outra se dá apenas pela vibração dos cordões vocais como o “p” e “b” em “pato” e “bato”, sendo então chamadas de homorgânicas. Dá-se esta definição aos fonemas cuja pronúncia depende do comportamento do(s) órgão(s) do aparelho fonador no momento da passagem do ar.

No português, ainda existem duas semivogais “i” e “u”. Estas tem composições articulatórias e acústicas que se assemelham às vogais, porém, não desempenham papel de núcleo da sílaba, devendo estar associadas a vogais para constituir uma sílaba. Outras característica é que as semivogais não são acentuadas. Existem alguns casos onde uma semivogal pode ser representada por um “e” ou um “o”como nas palavras mãe e pão.

### 2.1.4 Grafemas

A representação visual dos sons e contraparte escrita do fonema dá-se o nome de **grafema**, sendo este também a unidade mínima da escrita. Mínima, porque esta não pode ser desmembrada em dois ou mais sinais que também possam ser tratados como grafema [35]. Por exemplo, p, b e m são grafemas dos fonemas /p/, /b/ e /m/.

No português, existem 78 grafemas, diferenciados entre minúsculos e maiúsculos, já que as suas funções não podem ser aglutinadas. Dentre as classificações de grafemas, existem os fonológicos, onde estão presentes sinais ortográficos (“ ”. , ; : ? ! ... ( ) ? - ?). Estes podem agregar funções diversas na língua, como a alteração da entonação de uma sentença <!>. Existem ainda os ideogramas, como os números, as siglas e alguns símbolos ([ ] { } % * @ & + / º $ §).

Na língua portuguesa, há situações onde grafemas podem representar fonemas. Entretanto, em muitos casos, um grafema não corresponde a apenas um fonema e vice-versa. Dentro dessa equivalência entre grafemas e fonemas podem-se identificar as seguintes relações [25]:

- Um grafema para vários fonemas. Por exemplo, o sentido fonético encontrado no “c” das palavras coisa e césio, “k” e “s”, respectivamente.
- Um fonema para vários grafemas. Observa-se essa relação fortemente presente no fonema “s” presente na interpretação do “c”, “s”, “ss”, “ç”, “x” e “sc” em palavras como Cesário, nascer e expandir.
- Um grafema representa necessariamente um fonema. O grafema “j” é representado pelo fonema “j” somente, porém este fonema representa outros fonemas como o “g” em algumas ocorrências.
- Um fonema representa necessariamente um grafema. O fonema “r” é representado somente pelo grafema “r”, porém este grafema pode ser representado pelo fonema “R” como em raro.
- Relação biunívoca. Os grafemas “b”, “d”, “f”, “p”, “t” e “v” têm apenas uma representação fonológica, cuja contrapartida é igualmente estabelecida.
- Grafema mudo. No português, quando há a ocorrência do grafema “h” no início de palavras e o “u” em palavras como quitar e guincho, não possuem correspondentes fonéticos.
- Dígrafos. No português ocorre quando um fonema é representado por dois grafemas como em banho e chuva.
- Fonema cujo correspondente é somente um dígrafo.
- Dígrafo biunívoco. São os fonemas representantes dos dígrafos “nh” e “lh”.
- Dífono. Quando um grafema expressa dois fonemas, percebido principalmente quanto ao emprego do grafema “x” em palavras como fixar (“k s”) e ixofagia (“k z”).

### 2.1.5 Alfabeto fonético

Um **alfabeto fonético** é um sistemas de notação gráfica de sons, constituídos de símbolos e convenções que representam os sons [36]. Nestes sistemas, têm-se o cuidado para que cada som seja representado por um e um só símbolo, e que cada símbolo do alfabeto represente um único som da língua. Por exemplo, o símbolo <z> representa casos como **z**angado, e**x**ame e pe**s**o. Apesar de parecer simples, esta forma de notação contrasta com a relação entre fonemas-grafemas descrita anteriormente, fazendo necessário um tratamento especial para casos como o dos dígrafos, onde um único som pode ser representado por uma combinação de grafemas.

O *International Phonetic Alphabet* (IPA) - Alfabeto Fonético Internacional - [37] é um sistema de notação que começou a ser desenvolvido pela Associação Fonética Internacional, no final do século XIX, com o objetivo de descrever os sons do idioma falado. O IPA ainda guarda as informações referentes às posições e comportamentos dos órgãos articulatórios (posição da língua, abertura/fechamento da úvula, etc.) no momento em que determinado som é produzido [27]. O IPA também define que a sílaba tônica de uma palavra deve ser especificada com o símbolo <'>.

O *Speech Assessment Methods Phonetic Alphabet* (SAMPA) - Alfabeto Fonético dos Métodos de Avaliação da Fala - [38] é um alfabeto fonético baseado no IPA que usa caracteres ASCII de 7 bits. Começou a ser desenvolvido no final da década de 1980 pelo projeto ESPIRIT, com o objetivo de facilitar o processamento de intercâmbio de dados das transcrições na tecnologia da fonética e do discurso. Em sua primeira versão, continha apenas línguas presentes na comunidade europeia como o alemão e o francês, num total de seis idiomas. Já em 1993, o português, o grego e o espanhol foram acrescidos ao dicionário que hoje acumula todos os sons de aproximadamente 26 idiomas.

Este trabalho utiliza o alfabeto SAMPA para fazer o mapeamento entre fonemas e grafemas do PB. Por este motivo, são mostrados os fonemas do alfabeto SAMPA na Figura 2.3.

### 2.1.6 Transcrição fonética

**Transcrição fonética** é a representação gráfica dos sons produzidos usando os símbolos e as conversões de um alfabeto fonético escolhido [36]. É comum em uma transcrição fonética, os símbolos serem colocados entre colchetes < [] >. Em dicionários é muito comum ver a explicação da palavra acompanhada de um transcrição fonética que segue algum padrão. Por exemplo, utilizando os símbolos do SAMPA, transcrever-se-ia *casa* como ['kaza].

| Consoantes | | |
|---|---|---|
| **Oclusivas** | | |
| **Símbolo** | **Palavra** | **Transcrição** |
| p | pai | paj |
| b | barco | "barku |
| t | tenho | "teJu |
| d | doce | "dos@ |
| k | com | ko~ |
| g | grande | "gr6nd@ |
| **Fricativas** | | |
| f | falo | "falu |
| v | verde | "verd@ |
| s | céu | sEw |
| z | casa | "kaz6 |
| S | chapéu | S6"pEw |
| Z | jóia | "ZOj6 |
| **Nasais** | | |
| m | mar | mar |
| n | nada | "nad6 |
| J | vinho | "viJu |
| **Líquidas** | | |
| l | lanche | "l6nS@ |
| L | trabalho | tr6"baLu |
| r | caro | "karu |
| R | rua | "Ru6 |

| Vogais e ditongos | | | | |
|---|---|---|---|---|
| **Símbolo** | **Palavra** | **Transcrição** | **Palavra** | **Transcrição** |
| i | vinte | "vint@ | lápis | "lapiS |
| e | fazer | f6"zer | | |
| E | belo | "bElu | | |
| a | falo | "falu | | |
| 6 | cama | "k6m6 | madeira | m6"d6jr6 |
| O | ontem | "Ont6~j~ | | |
| o | lobo | "lobu | | |
| u | jus | ZuS | futuro | fu"turu |
| @ | felizes | f@"liz@S | | |
| i~ | fim | fi~ | | |
| e~ | emprego | e~"pregu (ou em-) | | |
| 6~ | irmã | ir"m6~ | | |
| o~ | bom | bo~ | | |
| u~ | um | u~ | | |
| aw | mau | maw etc.: iw, ew, Ew, (ow) | | |
| aj | mais | majS etc.: ej, Ej, Oj, oj, | | |
| 6~j~ | têm | t6~j~ etc.: e~j~, o~j~, u~j~ | | |

| Outros símbolos | |
|---|---|
| " | Acento tónico primário (posto antes da sílaba tónica) |
| % | Acento tónico secundário (posto antes da sílaba tónica) |
| . | Separador silábico |

Figura 2.3: Dicionário de fonemas SAMPA [38].

## 2.2 Sistemas texto-fala

A tarefa de um sistema TTS pode ser vista como o inverso de um reconhecedor de voz - um sistema de informação capaz de gerar voz parecida com fala humana a partir de um texto. Na comunidade acadêmica, o conceito de sistema TTS é comumente confundido com síntese de voz.

Thierry Dutoit, em [2], ressalta que os sistemas TTS não devem ser confundidos com outras máquinas falantes, tais como sistemas de resposta de voz que geram fala artificial através da concatenação de palavras isoladas ou partes de sentenças. Estes sistemas estão limitados a um certo vocabulário (geralmente 100 palavras) e as sentenças são pronunciadas

em uma estrutura bem restrita, como anúncios de chegadas de trens em estações, por exemplo. Em um sistema TTS, é impossível (e inútil) armazenar todas as palavras que serão ditas, pois o intuito desses sistemas é realmente gerar novas sentenças.

Tais sistemas possuem uma variedade enorme de aplicações e sua primeira utilização real estava como solução de leitura para cegos, onde o programa lia texto de um livro e o convertia em áudio. De início, estes sistemas eram muito mecânicos, mas a sua adoção por pessoas cegas foi surpreendente, onde as outras opções - leitura em braile e por outra pessoa real - não eram possíveis. Tais sistemas evoluíram e, hoje, facilitam a interação homem-máquina não só para cegos.

O principal limitante no uso de um sistema TTS é a qualidade. Ainda que tenham sido aprimorados, na maioria dos casos, não é necessário ser especialista para identificar uma voz sintetizada. Depois da adoção destes sistemas em alguns meios, não foi necessário muito tempo para perceber (particularmente no meio comercial) que os usuários se irritam e sentem-se desconfortáveis quando ouvem uma voz mecanizada. A experiência mostrou que os usuários preferem falas mais naturais.

Assim, o objetivo é construir um sistema capaz de (1) ler claramente um texto e (2) fazê-lo de forma humana. Dentro do meio acadêmico, estes requisitos são referidos como **inteligibilidade** e **naturalidade** [3]. É lógico que a comunicação humana falada também é, muitas vezes, acrescida com gestos faciais e manuais, o que, enriquece a relação. Porém, tais recursos não podem ser considerados na imitação da fala, pois não podem ser acrescidas a programas de computador. Existe também as diferenças entre a língua falada e a escrita. Quando escreve-se, não é possível expressar todos os sentimentos de maneira integral. Então, é normal se esperar que ao ler determinado texto, não consiga-se captar além do que foi escrito.

Porém, existe um ramo de estudo dentro de síntese de voz que cuida de incrementar a fala com emoções - medo, raiva, felicidade, tristeza, etc. Esta conhecida como **síntese de voz emotiva** ou **emocional** [3]. Embora não seja tão antiga quanto a irmã mais velha, esta área teve seus primeiros trabalhos publicados já no final da década de 90 [39–41]. As técnicas eram aplicadas em sintetizadores de formantes e chegaram a ser utilizadas comercialmente no DECTalk [42], sintetizador da Fonix.

### 2.2.1 Arquitetura

Depois de introduzidos os conceitos sobre voz e algumas considerações acerca de sistemas TTS, é hora de entender como o processo é feito. Começando pela forma de organização, a arquitetura mais utilizada é composta por dois componentes essenciais [43]: **módulo de**

**análise do texto** e **motor de síntese**. Em [3], Taylor denomina este modelo de **forma comum**.

A Figura 2.4 representa um diagrama funcional simples de um sistema TTS genérico. Nesta representação, o sistema possui um módulo de Processamento de Linguagem Natural (PLN) que é capaz de produzir uma transcrição fonética do texto a ser lido juntamente com a entonação e ritmo desejados. O outro módulo do sistema é o de Processamento Digital de Sinais (PDS) e, que transforma a informação simbólica que recebe em voz cujo som soa próximo do natural. PLN e PDS são, respectivamente, módulo de análise do texto e motor de síntese, e assim serão chamados no decorrer deste trabalho.

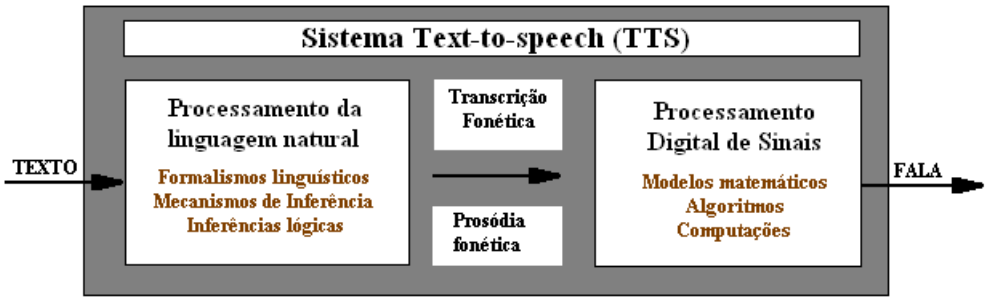


Figura 2.4: Diagrama funcional simples de um sistema TTS.

O formalismo particular escolhido para representar a informação intermediária do sistema TTS varia de um sintetizador para outro. Em [44], é apresentado um esquema que torna possível a interconexão de sistemas texto-fala diferentes através de linguagens baseadas em XML para representação interna dos dados. A intenção é fazer que módulos de diferentes sistemas sejam portáveis para outras aplicações. A plataforma MARY, que serve de base para o TTS deste trabalho, utiliza sua própria linguagem XML interna (MARYXML) [45].

#### 2.2.1.1 Análise do texto

O módulo responsável pela análise do texto - também chamado de *front end* - tem duas tarefas majoritárias. Primeiro, deve converter todo o texto de alto nível, que podem ser números, símbolos, abreviações e siglas, em formas equivalentes e por extenso. Na literatura, esta etapa é chamada de **normalização de texto**, ainda podendo ser referida com o pre-processamento de texto [27].

É importante fazer esta transformação inicial, porque os módulos posteriores entendem um número reduzido de caracteres, geralmente restritos somente ao alfabeto da língua. Para facilitar o entendimento, considera-se que o usuário queira sintetizar o seguinte trecho do exemplo 2.1:

O BACEN abriu 189 vagas, com salários de até R$ 1.000,00 (2.1)

Apesar de ser um texto bastante comum, um sistema TTS não pode trabalhar com todos os caracteres presentes. Os numerais e o símbolo *$* devem ser previamente tratados. Além destas questões, ainda existe as diversas possibilidades de leitura do texto. O acrônimo *BACEN* pode ser lido propriamente como *bacen* ou também *banco central*. Todos estes pontos são responsabilidade dos módulos de análise de texto. Uma forma normalizada do exemplo 2.1 poderia ser igual ao texto do exemplo 2.2:

o bacen abriu cento e oitenta e nove vagas com salários de até mil reais (2.2)

Converter de caixa alta para baixa todas as letras, remover sinais de pontuação, expandir abreviações e traduzir símbolos são trabalhos comuns aqui. Porém, é importante guardar a forma original do texto, pois etapas posteriores ainda podem precisar de informações que foram apagadas com a saída descaracterizada pelo módulo de análise. Por exemplo, para a identificação de uma sigla, um dos pré-requisitos é que a palavra analisada esteja toda em caixa alta. Um problema ocorre caso o normalizador de texto já tenha convertido para caixa baixa todo o texto antes do trabalho do conversor de siglas, a informação essencial sobre as siglas será perdida e o sistema não conseguirá identificar os *tokens* que deve transformar.

Alguns módulos de análise de texto também podem incluir a etapa de **etiquetamento gramatical** ou **analisador morfossintático**. Na academia, esta rotina é muito referenciada como *POS tagger* [46] e tem como objetivo extrair do texto original informações de ordem morfossintática. Estas podem ser utilizadas para fazer a desambiguação de homófonos (pronúncia igual e grafia diferente) e heterófonos (pronúncia diferente e grafia igual).

A segunda tarefa do *front end* pode ser dividida em três partes: atribuir transcrições fonéticas a cada palavra, dividí-las em unidades menores (sílabas) e determinar as partes (sílabas) tônicas. Estes processos são intitulados de **conversão grafema-fonema** (G2P), **silabação** e **marcação da sílaba tônica**, respectivamente. Todos eles irão fazer parte da representação linguística interna do sistema.

#### 2.2.1.2 Motor de síntese

O motor de síntese é quem produz a voz sintetizada. As entradas tradicionais deste módulo são as transcrições fonéticas (G2P) e informações prosódicas (silabação e tonicidade) advindas da análise de texto. Em sistemas TTS de alta qualidade, também adiciona-se o texto original etiquetado gramaticalmente, permitindo aumentar a naturalidade no resultado final.

O motor de síntese também é chamado de (*back end*). Dentre os diversos modelos já propostos para reproduzir a voz, deve-se destacar três [1]:

- **Síntese articulatória**: modelo baseado em regras, onde aparelho fonador e o processo de articulação ocorrido durante a fala são parametrizados [27] - velocidade e pressão do ar, abertura da boca, etc. Através de regras, estes modelos são modificados de modo a representar os fonemas. Embora seja bastante interessante representar o comportamento dinâmico do aparelho fonador, este método recebeu bem menos atenção que os baseados na análise de sinais e não teve o mesmo nível de sucesso.

- **Síntese por formantes**: é possível reproduzir voz fazendo uma fonte gerar formas de ondas periódicas, que serão filtradas e ressoadas por vários módulos em cascata ou paralelo. Na prática, este tipo de síntese faz o controle do *pitch* de e outras frequências do som. Este modelo também é conhecido como fonte-filtro. O sintetizador por formantes mais famoso é o de Klatt [47], inclusive usado no DECTalk. A naturalidade é o ponto fraco dessa abordagem, pois é muito difícil capturar todos os sutilezas envolvidas no processo, fazendo com que a voz fique "metalizada".

- **Síntese concatenativa**: de longe, é a forma de síntese que, até agora, atingiu maior apelo comercial. Nela, a partir de uma base de dados de áudio (geralmente extensa), vários segmentos de voz são rotulados e classificados para posteriormente serem concatenados e reproduzirem um novo áudio. A grande problema é trabalhar justamente com grandes *corpura*. Além de ser muito oneroso, estes sofrem com falta de flexibilidade, sendo necessário aplicar algumas técnicas para a modificação da prosódia [1].

Das três acima, somente a síntese concatenativa ainda permanece como estado da arte, porém, hoje já divide atenção com um abordagem mais nova e com alto potencial: a **síntese baseada em modelos escondidos de Markov** (HMM). O modelo não chega a ser tão novo, contudo, somente no final de década de 90 começou a se difundir, com os trabalhos de pesquisadores japoneses [48–51]. Não por acaso, alguns referenciam esta técnica como a síntese dos japoneses. Existe ainda uma denominação mais usual - síntese HTS.

A verdade é que muito antes de ser usada nos sintetizadores, as HMMs já eram amplamente difundidas na comunidade como excelentes ferramentas para reconhecimento de voz, sendo os estudos direcionados com o trabalho de Rabiner em [29]. Entretanto, ainda não se tinha uma forma de fazer as cadeias gerarem parâmetros acústicos de voz de maneira eficiente. Em [52], Tokuda, baseado em trabalhos anteriores [49, 50], apresentou o esquema principal para geração de parâmetros acústicos a partir de HMMs. Ainda que melhorias e no-

vas técnicas tenham sido agregadas, a estrutura inicial proposta por Tokuda ainda se mantém como a padrão. A Figura 2.5 ilustra o esquema.

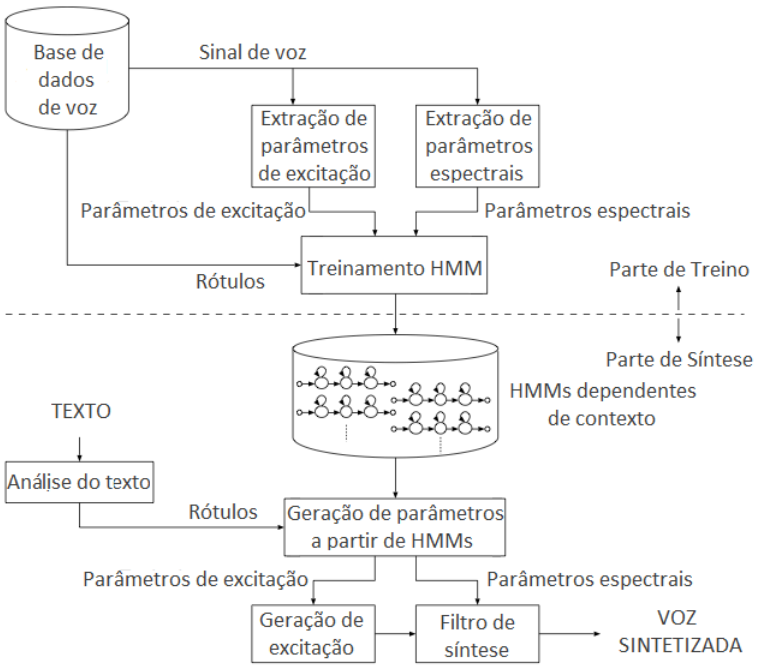


Figura 2.5: Esquema de sistema de síntese baseado em HMM. Adaptada de [52]

Em [53], os autores aplicam o modelo em uma língua específica - o inglês. Além do artigo, foram disponibilizadas ferramentas, códigos demonstrativos e recursos [13] como forma de promover o uso de HMM na área. Os programas desenvolvidos foram todos modificações e adaptações de uma ferramenta amplamente utilizada em reconhecimento de voz, o ***H****idden Markov Model* ***T****ool****K****it* (HTK) [54]. A sigla HTS (***H****idden Markov Model* ***T****oolkit for* ***S****ynthesis*) nasceu com um reconhecimento pelo empréstimo dos recursos já implementados pelo HTK e reaproveitados para síntese. Hoje, tais recursos servem como base para implementações de HTS em várias línguas [55–59], inclusive o PB, que teve a solução adaptada por Maia [12]. É importante ressaltar que apesar de Maia ter aplicado a técnica ao PB, o produto final não constitui um sistema TTS completo, pois várias etapas da análise de texto foram suprimidas, mantendo o foco somente na explicação da técnica. Neste trabalho, propõem-se preencher estas lacunas e ir mais além.

Como HTS é a abordagem empregada para o problema em questão, separou-se uma seção para explicar melhor o assunto.

### 2.2.2 Síntese baseada em HMM.

Uma das vantagens de usar modelos escondidos de Makov é que a síntese se torna um processo treinável, portanto, mais flexível que a abordagem concatenativa. A flexibilidade implica na possibilidade de se realizar alterações na voz sem recorrer à grandes bases de dados [60–62]. De acordo com [63], outra vantagem seria a obtenção de voz com aplicabilidade[1] através de bases pequenas - em torno de 80 sentenças. Além disso, sintetizadores baseados em HMM podem competir em qualidade com sintetizadores concatenativos [64, 65] usando pouco mais de uma hora de áudio.

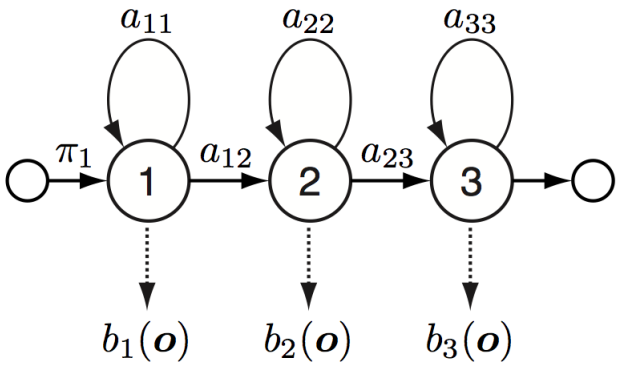


Figura 2.6: Ilustração de um cadeia de Markov.

A Figura 2.6 apresenta o esquema simples de uma cadeia de Markov utilizada nas técnicas de reconhecimento e síntese de voz. A técnica é divida em duas etapas: parte de treino e de síntese.

#### 2.2.2.1 Treino

Começando com o processo de treinamento ainda pode-se subdividi-lo em (1) extração dos parâmetros acústicos, (2) geração dos rótulos contextuais e (3) treinamento dos HMMs.

**Parâmetros acústicos**: extrair os parâmetros acústicos significa utilizar as ferramentas de processamento de sinais para retirar informações do som. Como foi dito em capítulos anteriores, as amostras do sinal no domínio do tempo não são utilizadas diretamente. Os parâmetros acústicos podem ser vários, porém, os mais utilizados são a frequência fundamental (*pitch*), os coeficientes cepstrais da escala mel (MFCC) [30] e parâmetros não-periódicos [66].

**Rótulos contextuais**: esta etapa utiliza as informações sobre a sentença, que são passadas pelo módulo de análise de texto (conversão grafema-fonema, silabação, determinação

[1]Significa que poderia ser utilizada em algumas aplicações.

da sílaba tônica e etiquetamento gramatical), para criar os rótulos contextuais. Tais rótulos captarão todas estas relações em torno do fonema e servirão para realizar uma reprodução mais natural da prosódia. Estas informações estão categorizadas nos mais diferentes níveis. Abaixo, uma lista não-exaustiva das informações armazenadas em diferentes níveis.

- Nível de fonema: fonema anterior, atual, posterior, posição do fonema na sílaba, etc.
- Nível de sílaba: sílaba tônica ou não, número de fonemas da sílaba anterior, atual, posterior, posição da sílaba atual dentro da palavra, etc.
- Nível de palavra: etiqueta gramatical da palavra anterior, atual, posterior, número de sílabas da palavra anterior, atual, posterior, etc.
- Nível de frase: número de sílabas, palavras precedendo, sucedendo à frase, posição da frase atual na sentença.
- Nível de sentença: número de sílabas, palavras, frases na sentença.

A determinação de que informações os rótulos contextuais deverão conter são baseadas nas prosódias características e suposições linguísticas do idioma em questão. Para o PB, ainda é utilizado no modelo adotado pelo inglês neste trabalho. Desenvolver um padrão adaptado para o PB é de extremo interesse, contudo, este não pode ser confeccionado sem a ajuda de estudiosos da fonética e fonologia.

**Treinamento das HMMs**: as HMM's utilizadas correspondem a uma cadeia de $S$ estados do tipo *left-to-right*, com $S = 5$. A saída de cada HMM, $o^i$, correspondente ao $i$-ésimo frame, pode ser dividida em cinco partes (*streams*),$o^i = [{o_1^i}^T ... \ {o_5^i}^T]^T$, como ilustrado na Figura 2.7. Cada *stream* guarda as seguintes informações:

- *Stream* 1 ($o_1^i$): vetor composto dos coeficientes mel-cepstrais, $\{c_0^i, ... \ , c_M^i\}$, primeira derivada (delta), $\{\Delta c_0^i, ... \ , \Delta c_M^i\}$, e segunda derivada (delta-delta), $\{\Delta^2 c_0^i, ... \ , \Delta^2 c_M^i\}$;
- *Streams* 2, 3, 4 ($o_2^i, o_3^i, o_4^i$): composto pelo logaritmo da frequência fundamental, $\log(F0^i)$, e respectivos delta, $\Delta \log(F0^i)$, e delta-delta, $\Delta^2 \log(F0^i)$;
- *Stream* 5 ($o_5^i$): vetor composto pelos coeficientes não-periódicos, $\{b_1^i, ... \ , b_5^i\}$, e respectivas primeira e segunda derivadas, $\{\Delta b_1^i, ... \ , \Delta b_5^i\}$ e $\{\Delta^2 b_1^i, ... \ , \Delta^2 b_5^i\}$;

Para cada HMM $k$, a duração dos $S$ estados são consideradas como vetores, $d^k = \left[d_1^k ... \ d_S^k\right]^T$, onde $d_s^k$ representa a duração do estado $s$. Os vetores de duração, $\left\{d^1 ... \ d^K\right\}$, onde $K$ é o numero total de HMM's que representam a base de dados, são modelados por

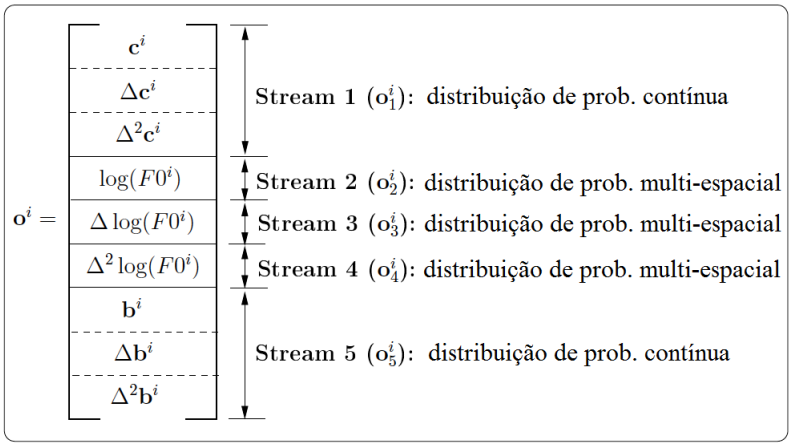


Figura 2.7: Estrutura do vetor de saída $o^i$: coeficientes mel-cepstrais da escala mel, $c^i$, frequência fundamental, $\log(F0^i)$, e os parâmetros aperiódicos, $b^i$. Adaptada de [12]

uma distribuição de misturas de gaussianas simples com $S$ dimensões. A utilização de HMM's com durações de estados explícitas vem do conceito de HSMM (*Hidden Semi-Markov Models*). Em [67], Heiga explica a aplicação e funcionamento deste modelo para a síntese de voz.

Durante a etapa de treinamento, é utilizada uma técnica chamada de agrupamento por contexto. Nela, os *streams* de MFCC, F0, parâmetros não-periódicos e modelos de duração são separadas de acordo com as informações captadas pelos rótulos contextuais. No fim do processo, $3S + 1$ diferentes árvores de decisão são geradas, sendo $S$ árvores para MFCC (uma para cada estado $s$), $S$ árvores para a frequência fundamental (uma para cada estado $s$) e $S$ árvores para os parâmetros não-periódicos (uma para cada estado $s$) e uma para árvore para a duração dos estados. As folhas destas árvores correspondem às *streams* das HMM's treinadas e serão selecionadas durante o processo de síntese.

#### 2.2.2.2 Síntese

O processo de sintetizar uma sentença é conduzido através das etapas de (1) geração dos rótulos fonéticos, (2) seleção e concatenação das HMM's, (3) determinação dos parâmetros e (4) construção da excitação e filtragem.

**Geração dos rótulos contextuais**: o processo de síntese começa com a conversão das informações sobre a sentença em rótulos contextuais, da mesma forma que aconteceu no treino.

**Seleção e concatenação das HMM**: na síntese, estes rótulos são utilizados para selecionar as folhas das $3S + 1$ árvores - criadas durante o treinamento - e que serão utiliza-

das para sintetizar o texto de entrada. No fim do processo, quatro sequências (MFCC, F0, aperiódicos e duração) de uma HMM lógica são formados a partir das folhas selecionadas das árvores de decisão. As sequências que correspondem à MFCC, $F0$ e coeficientes não-periódicos irão compor a HMM que reproduzirá a voz e a prosódia irá ser controlada pela HMM que modelará a duração do fonema. O procedimento é aplicado a cada fonema da sentença de forma a criar um *array* de HMM's.

**Determinação dos parâmetros**: para gerar os parâmetros acústicos, primeiramente, as HMM's de duração são utilizadas, definindo a sequência de estados $s = \{s_1, ... , s_L\}$, com $L$ sendo o número de frames da sentença a ser sintetizada e $s_i$ o estado da HMM de onde o frame receberá os valores. Após isto, os coeficientes mel-cepstrais, logaritmos da frequência fundamental e coeficientes não-periódicos são determinados a partir de cada HMM correspondente de forma a maximizar suas probabilidades de saída dado $s$, levando em consideração os valores de delta e delta-delta, de acordo com o algoritmo proposto por Toda em [68].

**Construção da excitação e filtragem**: a última fronteira que falta para realizar a síntese é utilizar os parâmetros obtidos na etapa anterior para produzir o sinal de voz. Este pode ser dividido em duas partes. A primeira corresponde em construir a excitação do sinal usando os logaritmos da frequência fundamental e os coeficientes não-periódicos com um método de *vocoding* de alta-qualidade [66]. A segunda parte é utilizar os MFCC's para configurar o filtro MLSA (*Mel Log Spectrum Approximation*) [30].

As duas técnicas mencionadas no parágrafo anterior (*vocoding* e filtro MLSA) são bastante refinadas e detalhá-las está fora do escopo deste trabalho. Entretanto, é possível fazer uso de uma simplificação que mostre a relação dos métodos para a geração da voz. O primeiro método (*vocoding*) utiliza a frequência fundamental e os coeficientes não-periódicos para gerar uma forma de onde inicial, que possui características ligadas aos parâmetros de entrada. Por exemplo, na Figura 2.8, é mostrado que a saída do *vocoder* é um trem de impulsos (1) com período entre as amostras igual a $T0$, sendo $T0 = 1/F0$. Este sinal primário será então remodelado de acordo com um filtro (MLSA), que é configurado com os coeficientes mel-cepstrais. O resultado é o sinal de voz (2).

A analogia se assemelha muito com o modelo fonte-filtro utilizado na síntese por formantes [47], onde o *vocoder* faz o papel de fonte e o MLSA o de filtro. É que lógico de diversos conceitos deixaram de ser expostos, porém, a comparação permite verificar como as duas técnicas são utilizadas. Um ponto interessante é que a extração de MFCC's é uma técnica normalmente não-inversível (com perda de informação do sinal), não sendo possível reconstruí-lo. Acontece que os parâmetros acústicos de excitação (F0 e aperiódicos), conseguem suprir estas deficiências e casados com o MFCC possibilitam a reconstrução do sinal.

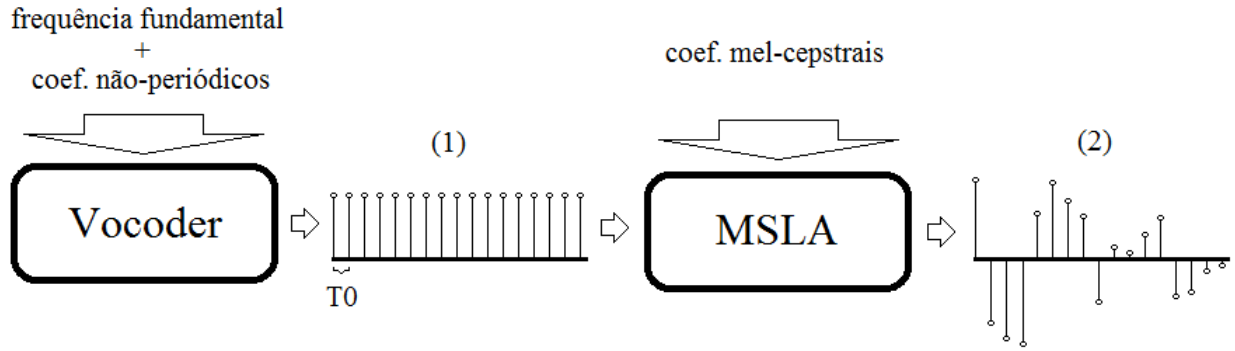


Figura 2.8: Relação entre as técnicas de *vocoding* e MLSA para a produção de voz.

A Figura 2.9 é uma boa representação de como cada estado das HMM's pode gerar vários *frames* de voz. Inicialmente, cada fonema é modelado por uma cadeia de estados e cada um destes estados pode gerar um ou mais *frames*. Por conseguinte, a quantidade de *frames* dos estados é determinada pelo modelo de duração incorporado no HMM.

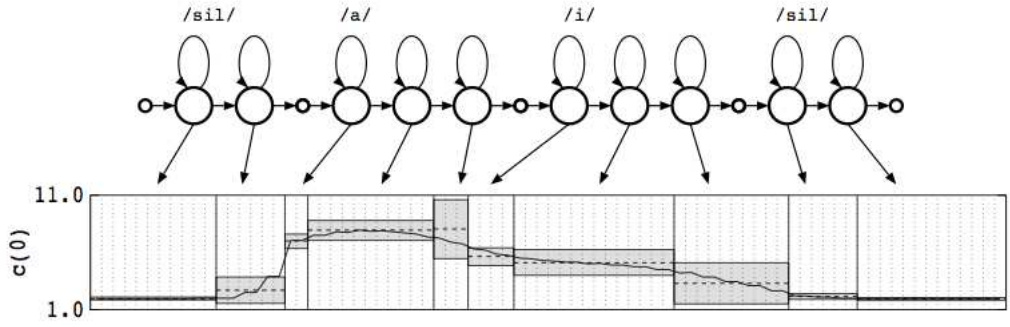


Figura 2.9:

# Capítulo 3

# Algoritmos para a análise do texto

Neste capítulo serão abordados os algoritmos desenvolvidos para a etapa de análise do texto (*front end*). Os processos desta etapa têm a função de realizar a conversão de um texto de linguagem de alto nível, geralmente contendo siglas, abreviaturas, símbolos e numerais, para um formato de mais baixo nível e que possa ser tratado (entendido) por módulos posteriores do sistema. Números devem ser estendidos, símbolos convertidos, caracteres especiais tratados, etc.

O mais comum é que várias destas tarefas sejam realizadas com regras e dicionários, onde somente se verifica a ocorrência do caso e aplica-se a conversão procurando diretamente em uma tabela ou lista. Porém, existem casos mais difíceis de serem convertidos, como os numerais, que podem ser interpretados de formas diferentes, onde até o contexto interfere na interpretação.

Todos os componentes e algoritmos descritos nas seções subsequentes são fortemente dependentes da língua. Neste capítulo, apresentaremos os dicionários e algoritmos de conversão propostos para resolver os problemas inerentes à tarefa de normalização e conversão de texto, sendo que estes só terão aplicabilidade para o português do Brasil.

## 3.1 Visão geral

O processo de análise do texto pode ser dividido em duas partes, sendo a primeira chamada de **normalização de texto**, **pre-processamento do texto** ou **processamento de linguagem natural** (PLN) [2,3,69]. Nesta etapa, a tarefa consiste em somente transformar textos de linguagem de alto nível (siglas, abreviaturas, símbolos e numerais) para um formato com um conjunto limitado de grafemas. Feita esta transformação, dá-se início a segunda etapa,

referida como **análise fonética** [1]. Nela, são realizadas as transcrições fonéticas para cada palavra, dividindo-as e determinando as unidades tônicas.

Os módulos da segunda etapa geralmente são soluções baseados em regras específicas para uma determinada língua. Devido a isto, estes módulos devem trabalhar com um conjunto reduzido de grafemas na entrada - um domínio específico. Este conjunto geralmente é o alfabeto fonético em questão. É aqui que nasce a importância dos módulos PLN, pois os textos na entrada de um sistema TTS geralmente (quase sempre) não possuem somente letras do alfabeto. Numerais e símbolos são alguns exemplos de grafemas frequentemente usados na escrita. Portanto, o PLN tem a tarefa de normalizar os textos para um domínio compreendido pelo computador.

Na Seção 3.2, são apresentados os algoritmos referentes ao módulo PLN do sistema TTS. Prosseguindo na Seção 3.3, são ilustrados os algoritmos que compõem o módulo para análise fonética. Todos foram implementados na linguagem de programação Java e estão disponíveis para *download* gratuito em [5]. Optou-se pela linguagem Java, por ser uma solução de boa portabilidade e de bom conhecimento por parte do autor deste trabalho.

## 3.2 Processamento de linguagem natural

O objetivo do módulo PLN é fornecer aos computadores a capacidade de entender textos. Na verdade, o termo processamento de linguagem natural inclui um conjunto bem amplo de questões, como interpretar contextos, aprender conceitos e até extrair ideias complexas de um texto. Entretanto, no sistema TTS deste trabalho, assim como nos mais atuais, as ferramentas PLN têm um propósito mais simples: reconhecer padrões e fazer com que os módulos posteriores possam processar as informações passadas na entrada.

### 3.2.1 Separador de frases

Um dos primeiros algoritmos na etapa de normalização do texto, consiste somente em separar as frases do texto de acordo com os sinais de pontuação que indiquem final de frase ou período [34], como ponto <.> , ponto de exclamação <!>, ponto de interrogação <?> e reticências <...>. Apesar de simples, muito cuidado deve se tomar com as ocorrências do ponto, pois também podem indicar abreviaturas, como em <Av.> → Avenida e <Sr.> → Senhor, e formatação de determinados numerais, como valores monetários.

Para fins de implementação, primeiro faz-se uma verificação com o expansor de abre-

viaturas, descrito em 3.2.3, e de numerais, na Seção 3.2.6, para averiguar se existem casos a serem tratados. O processo é necessário para evitar que estas ocorrências venham a fazer com que o separador de frases trabalhe de forma incorreta. Por exemplo, o texto "O eng.° deve R$ 2.100,92 ao conselho." possui somente uma frase, apesar de existirem três sinais de pontuação.

O separador de frases pode ser considerado um módulo em anexo a todos os outros, pois quando deseja-se varrer um texto, sempre existe a necessidade de fazer a segmentação do mesmo para posteriormente analisá-lo.

### 3.2.2 Separador de palavras

Nesta etapa, busca-se dividir a frases em palavras que foram passadas pelo separador de frases. Na língua portuguesa, as palavras são separadas com o espaço em branco, facilitando a tarefa. Entretanto, há de se levar em consideração que existem palavras que são faladas separadamente e não seguem a afirmativa anterior. Estes casos podem ser subdivididos em três grupos:

- Palavras compostas: guarda-chuva, caminhão-pipa, chapéu-de-sol, abaixo-assinado, quebra-nozes, carro-forte, decreto-lei, pica-pau, manda-chuva, saca-rolhas, porta-voz.
- Palavras com sufixos: ex-comunista, anti-aéreo, pós-graduação, sub-região, co-autor,
- Formas verbais com mesóclise e ênclise: convidar-me-ão, entregar-lhe, concedê-lo, observar-me-á, faz-se-ão.

Apesar de muitas destas palavras só manterem o sentido correto quando estão juntas, estas são tratadas como palavras separadas pelos módulos posteriores. Mesmo o caso de colocação pronomial, que se subordina diretamente às formas verbais, será tratado de maneira independente, não afetando o desempenho do sistema.

Assim como o módulo apresentado na Seção 3.2.1, o separador de frases também pode ser considerado como integrante dos demais componentes.

### 3.2.3 Expansor de abreviaturas

Na língua portuguesa, abreviaturas são palavras truncadas e terminadas por ponto. Para fins de implementação, a conversão de abreviaturas é a primeira etapa a ser feita, pois,

como foi explicado na seção 3.2.1, diminui as chances do separador de frases operar erroneamente. A abordagem proposta é a utilização de dicionário para armazenar os casos mais recorrentes na língua portuguesa. O dicionário atual do sistema contém 50 ocorrências e pode ser facilmente incrementado de acordo com a necessidade.

Tabela 3.1: Exemplos de abreviaturas e as respectivas expansões.

| Abreviatura | Expansão | Abreviatura | Expansão |
|---|---|---|---|
| V.A. | vossa Alteza | VV.AA. | vossas altezas |
| V. Ex.ª | vossa excelência | V. Mag.ª | vossa magnificência |
| V. S.ª | vossa senhoria | V. S. | vossa santidade |
| V. M. | Vossa Majestade | V. A. R. & I. | vossa alteza real e imperial |
| V. A. I. | vossa alteza imperial | V. A. R. | vossa alteza real |
| V. A. Ilmª. | vossa alteza ilustríssima | Hon. | honorável |
| Il.mo | ilustríssimo | v. | você |
| Prof. | professor | prof. | professor |
| Prof.ª | professora | prof.ª | professora |
| Sr. | senhor | Srs. | senhores |
| sr. | senhor | srs. | senhores |
| Sra. | senhora | Sras. | senhoras |
| sra. | senhora | sras. | senhoras |
| Dr. | doutor | dr. | doutor |
| jan. | janeiro | fev. | fevereiro |
| mar. | março | abr. | abril |
| maio | maio | jun. | junho |
| jul. | julho | ago. | agosto |
| set. | setembro | out. | outubro |
| nov. | novembro | dez. | dezembro |

Exitem certas abreviaturas que se confundem com outros casos, necessitando de desambiguação por contexto, que ainda não é considerada no trabalho. É o caso do texto <dez.>, que pode vir a ser a abreviação de dezembro, como pode ser visto na Tabela 3.1, ou o numeral <10> por extenso no final de uma frase.

Aqui, há de se mencionar um problema. Muitas vezes por questões de economia de tempo ou espaço, as pessoas abreviam levianamente as palavras, fazendo com que esta tarefa se torne ainda mais problemática. Na língua portuguesa, também não existe uma regra geral, que englobe todos os casos quando se tem a necessidade de abreviar. Exitem ainda situações onde as abreviações nascem espontaneamente e sem respeitar nenhum critério, como é caso da Internet. Abreviaturas como <vc> → você e <hj> → hoje vieram da necessidade de uma comunicação rápida e instantânea, características da rede mundial de computadores. Assim como estes casos, outros podem ser restritos a certos grupos de pessoas, adicionando mais desafios na etapa.

Para o presente trabalho, este último problema será desconsiderado, por levar em consideração, que estas formas de escrita não advém de nenhuma norma ou regra da língua portuguesa. Apesar de não descartar a utilização do sistema TTS deste trabalho para ambientes como a Internet, há de se levar em conta que um provável usuário deve manter um mínimo de formalidade durante a sua utilização.

### 3.2.4 Conversor de símbolos

É muito comum a utilização de caracteres que representam muito mais do que um simples fonema. Tais caracteres, aqui definidos como símbolos, não são representados como um único som (fonemas), mas sim como a união de vários. Esta característica faz com que eles tenham que ser devidamente tratados.

Para o sistema TTS, símbolos são caracteres simples, que ocupam espaço único, não pertencentes ao alfabeto português e que não sejam números (cardinais, ordinais ou romanos) e nem sinais de pontuação. No geral, são relações matemáticas, letras gregas e caracteres especiais normalmente utilizados no dia-a-dia.

Existem ainda circunstâncias em que a leitura correta de um símbolo depende do contexto, como quando estes vêm acompanhados de um numeral. Deste modo, alguns símbolos podem ser lidos tanto no singular quanto no plural.

Tabela 3.2: Exemplos de como um símbolo pode variar dependendo dos grafemas anteriores

| Sigla | Conversão |
|---|---|
| Caso 01 | um **grau Celsius** |
| Caso 02 | dez **graus Celsius** |

Tal característica fez com que fosse feita uma divisão para o tratamento adequado: símbolos independentes de contexto e símbolos dependentes de contexto. A primeira lista

Tabela 3.3: Exemplos de símbolos independentes de contexto, juntamente com as respectivas conversões ortográficas

| Símbolo | Conversão ortográfica | Símbolo | Conversão ortográfica |
|---|---|---|---|
| $=$ | igual a | $\neq$ | diferente de |
| $+$ | mais | - | menos |
| $\div$ | dividido por | $\times$ | vezes |
| $\leq$ | menor ou igual | $\geq$ | maior ou igual |
| $\infty$ | infinito | * | asteristico |
| ˜ | til | $\emptyset$ | vazio |
| $\alpha$ | alpha | $\beta$ | beta |
| $\Gamma$ | gamma | $\gamma$ | gamma |
| $\Delta$ | delta | $\delta$ | delta |
| $\epsilon$ | epsilon | $\eta$ | eta |
| $\zeta$ | zeta | $\theta$ | teta |
| $\Lambda$ | lambda | $\lambda$ | lambda |
| $\pi$ | pi | $\Pi$ | pi |
| $\rho$ | ro | $\Sigma$ | sigma |
| $\psi$ | psi | $\omega$ | ômega |
| & | e comercial | @ | arroba |

(independentes) é composta por aqueles símbolos que não variam, podendo ser vistos na Tabela 3.3. O segundo rol de símbolos (dependentes), apresentados na Tabela 3.4, estão aqueles que dependem da verificação do caractere anterior para determinar como o símbolo será escrito, como apresentado no exemplo da Tabela 3.2. Assim como na Seção 3.2.3, as tabelas também podem ser aumentadas, sendo necessário somente obedecer a classificação quando ao contexto.

### 3.2.5 Leitor de siglas

A leitura de siglas também é uma etapa importante na normalização do texto. Tarefas constantemente presentes na língua escrita, a detecção e a leitura de siglas nem sempre são simples. A liberdade para a criação, assim como a quantidade de siglas no cotidiano, impossibilitam a normalização somente via dicionário, sendo necessário a criação de um abordagem

Tabela 3.4: Exemplos de símbolos dependentes de contexto e respectivas conversões ortográficas

| Símbolo | Conversões ortográficas | |
|---|---|---|
| ° | grau | graus |
| °F | grau Fahrenheit | graus Fahrenheit |
| J | Joule | Joules |
| W | Watt | Watts |

mais eficiente. O leitor de siglas deve ser capaz de identificar (detecção) e tratar (leitura) corretamente uma ocorrência.

O processo de detecção verifica se uma determinada palavra analisada pelo módulo PLN é realmente uma sigla. Para ser considerada como tal, a palavra deverá obedecer à três regras de detecção:

1. Devem sempre ter no mínimo duas letras. E quando for formada por somente duas letras, todas devem ser maiúsculas.
   - Ex.: PA - Estado do **Pa**rá.
   - Ex.: BR - **Br**asil
2. Admitisse que para siglas maiores ou iguais à três letras, uma e somente uma seja minúscula.
   - Ex.: UFPa - **U**niversidade **F**ederal do **Pa**rá.
   - Ex.: UnB - **Un**iversidade Federal de **B**rasília
3. As letras não podem vir separadas por espaços ou pontos.
   - Ex.: I.B.A.M.A. - **I**nstituto **B**rasileiro de **M**eio **A**mbiente.

Caso o módulo detecte a presença de uma sigla, o próximo passo é fazer uma análise para saber como a sigla deve ser lida, podendo gerar três possíveis resultados: (1) expandir, (2) soletrar ou (3) pronunciar. Por exemplo, (1) <BR> $\rightarrow$ *brasil*, (2) <CNPJ> $\rightarrow$ *c n p j* e (3) <EMBRAPA> $\rightarrow$ *embrapa*.

A Figura 3.1 é um fluxograma que ilustra como os dois procedimentos descritos nos parágrafos anteriores (detecção e leitura) são feitos. Trata-se do algoritmo implementado no leitor de siglas desta seção. A ideia inicial de um algoritmo que misture regras e dicionários

para tratar de siglas é de [70], onde é aplicado um procedimento similar para o português europeu.

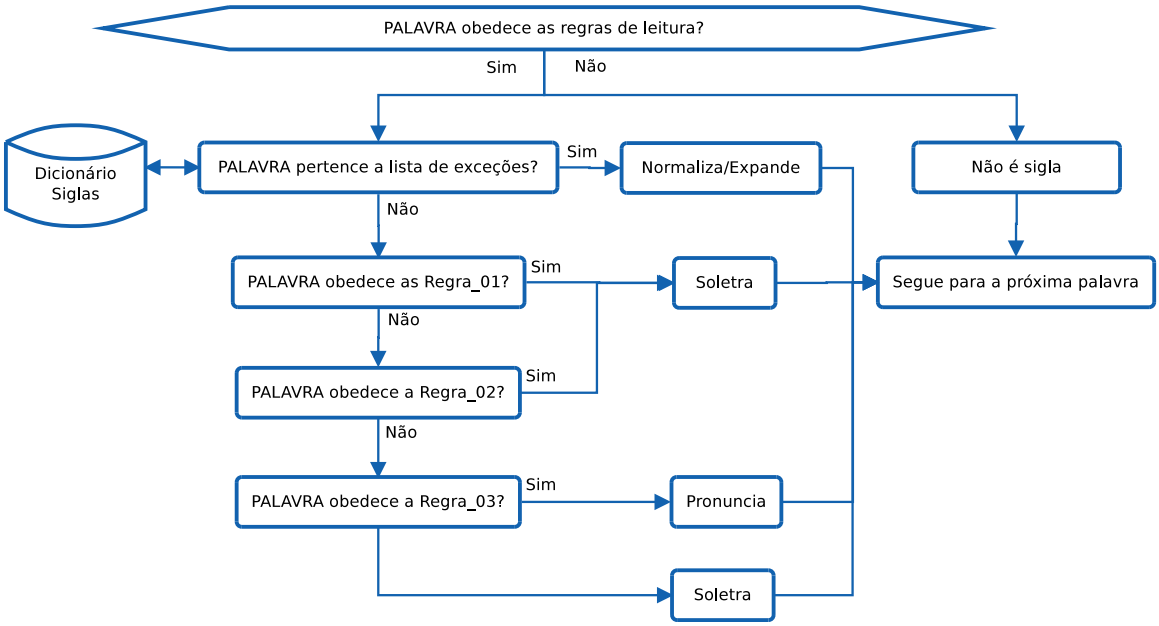


Figura 3.1: Fluxograma para o tratamento de siglas do sistema.

O primeiro passo do algoritmo é verificar se a palavra analisada é realmente uma sigla, comparando com as regras de detecção. Caso negativo, o módulo segue para o próxima palavra. Noutra circunstância (palavra é uma sigla), será feita a análise para decidir como a sigla será lida - expandir ou soletrar ou pronunciar. Esta análise é feita com a utilização de três regras de leitura e uma lista. Estas regras levam em consideração a forma de organização de vogais (V) e consoantes (C) dentro da siglas:

- Regra_01: Se a palavra for formada somente por consoantes, então a sigla é soletrada.
  - Ex.: RG, CPF, CNPJ, CBF.
- Regra_02: Siglas que estejam organizadas com VCC, VCCV, VCCC e CCV devem ser soletradas:
  - Ex.: VCC - IBM, FMI, UDP, FBI,USB.
  - Ex.: VCCV - IPTU, IPVA, UFPA.
  - Ex.: VCCC - UFRJ.
  - Ex.: CCV - NBA, CSA.
- Regra_03: Siglas que estejam organizadas com CVCV, VCVCVC, CVCVCV, VCVC, VCV, CVVC, CVV e CVC devem ser pronunciadas:

- Ex.: CVCV - NASA, SIDA.
- Ex.: VCVCVC - ANATEL.
- Ex.: CVCVCV - ANVISA.
- Ex.: VCVC - OTAN, IMAP, ENEM.
- Ex.: VCV - ONU, ITA,OSI.
- Ex.: CVVC - BIOS.
- Ex.: CVV - CIA, FIA.
- Ex.: CVC - FAB, CEP, SIS, SUS, DEM.

A lista (DicionarioSiglas) contém siglas que são exceções, pois ou são incompatíveis com as regras de leitura ou não existe modo de tratá-las por meio de regras. Por exemplo, uma sigla incompatível é HIV (*Human Immunodeficiency Virus*), porque obedece a regra de leitura Regra_03, porém não deve ser pronunciada, e sim, soletrada. Já IEEE, que é lida de modo excepcional, nem possui regra de leitura apropriada e acabaria também sendo soletrada. Se a sigla não estiver em DicionarioSiglas e nenhuma das regras de leitura for atendida, a sigla é soletrada. A Tabela 3.5 apresenta alguns exemplos da lista DicionarioSiglas.

Tabela 3.5: DicionarioSiglas: Lista com as siglas excepcionais.

| Sigla | Normalização |
|---|---|
| HIV | h i v |
| TI | t i |
| IEEE | i três é |
| USP | usp |
| MP3 | m p três |
| POP3 | pop três |
| EUA | estados unidos da américa |
| SPFC | são paulo futebol clube |
| TV | televisão |
| UE | união européia |

### 3.2.6 Expansor de numerais

Segundo [34], os numerais podem ser classificados em cardinais, ordinais, multiplicativos e fracionários, porém destes, os que mais necessitam de atenção são os dois primeiros, já que

são os mais utilizados na língua portuguesa. Devido a esta maior importância, somente estes dois são tratados dentro do módulo de expansão de numerais.

Sem dúvidas, o componente do módulo PLN mais complexo e trabalhoso é o expansor de numerais. Apesar de poderem ser divididos basicamente em cardinais e ordinais, os números podem estar inseridos em diversos contextos, podendo gerar diversas interpretações e leituras. O primeiro problema é descobrir qual é o tipo de numeral: ordinal ou cardinal. O segundo é saber em que contexto este está inserido.

O primeiro problema pode ser resolvido com a simples verificação do caractere junto ao numeral. Neste caso, os caracteres <ª> e <º> evidenciam a existência de um cardinal. O segundo é mais complicado e depende de uma análise caso a caso.

É importante frisar que os algoritmos propostos nesta seção fazem somente a conversão de números inteiros. Entretanto, a normalização de números racionais é possível, sendo necessário somente um tratamento anterior a expansão. Por exemplo, para estender o numeral <32,01>, seria preciso primeiro separar as frações para depois expandir individualmente. Apesar de parecer mais limitado, o procedimento de expandir somente números inteiros permite que o expansor seja utilizado da maneira mais adequada naquela situação. Um exemplo claro é quando se deseja expandir números fracionários, que dependendo do contexto podem gerar as mais diversas interpretações. Por exemplo, a seguinte frase: "O governo deve 2,5 bilhões.". Numa lista não-exaustiva, várias leituras diferentes poderiam ser feitas da mesma sentença:

- O governo deve dois vírgula cinco bilhões de reais.
- O governo deve dois bilhões e 500 milhões de reais.
- O governo deve dois e meio bilhões de reais.

Carregar todas essas diferentes interpretações para dentro no módulo de expansão seria muito trabalhoso e o limitaria. Por outro lado, fixar somente uma destas formas "engessaria"o expansor e o faria aceitar somente uma única interpretação que tivesse números não inteiros. Dessa forma, quem decide a normalização adequada é um algoritmo fora do expansor.

#### 3.2.6.1 Algoritmo de expansão de números cardinais

A conversão de números cardinais é feita tratando separadamente as unidades, dezenas, centenas, milhares, etc. A metodologia é composta por vários algoritmos menores que são responsáveis por converter algarismos de ordem específica. Entende-se ordem como a quantidade de dígitos válidos em um numeral, excluindo-se os zeros à esquerda por não influenciarem

na contabilização. Sendo assim, números com dígito único são tratados pelo algoritmo que normaliza unidades. Dois dígitos, transformados pelos métodos responsáveis por dezenas e assim em diante.

A abordagem é baseada em dicionários, onde verifica-se uma ocorrência e através de uma lista, retorna-se o resultado. É nessas listas que se encontram a extensão dos números cardinais. Ao todo, existem cinco listas. As Tabelas 3.6, 3.7, 3.8 e 3.9 exibem quatro destas. A lista empregada na transformação de unidade de milhar não foi exposta por ser muito similar à da Tabela 3.6. A diferença reside na ausência das duas primeiras linhas. Onde não se utiliza *zero mil* e nem *um mil*. A terceira coluna das tabelas apresenta os fonemas correspondentes ao numeral normalizado.

Tabela 3.6: Lista das extensões das unidades

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 0 | zero | zeru |
| 1 | um | u˜ |
| 2 | dois | dojs |
| 3 | três | trejs |
| 4 | quatro | kwatru |
| 5 | cinco | si˜ku |
| 6 | seis | sejs |
| 7 | sete | setSi |
| 8 | oito | ojtu |
| 9 | nove | novi |

Tabela 3.7: Lista das extensões das dezenas entre 10 e 19

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 10 | dez | dejs |
| 11 | onze | o˜zi |
| 12 | doze | dozi |
| 13 | treze | trezi |
| 14 | quatorze | kwatoXzi |
| 15 | quinze | ki˜zi |
| 16 | dezesseis | dezesejs |
| 17 | dezessete | dezesetSi |
| 18 | dezoito | dezojtu |
| 19 | dezenove | dezenovi |

O algoritmo principal tem o papel fundamental de identificar a quantidade de algarismo e repassar à rotina responsável. Procedimentos que manipulam números maiores utilizam diretamente métodos que estão abaixo deles (métodos que manipulam numerais de menor

Tabela 3.8: Lista das extensões das dezenas

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 20 | vinte | vi~tSi |
| 30 | trinta | tri~ta |
| 40 | quarenta | kware~ta |
| 50 | cinquenta | si~ke~ta |
| 60 | sessenta | sese~ta |
| 70 | setenta | sete~ta |
| 80 | oitenta | ojte~ta |
| 90 | noventa | nove~ta |

Tabela 3.9: Lista das extensões das centenas

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 100 | duzentos | se~j~ |
| 200 | duzentos | duze~tus |
| 300 | trezentos | treze~tus |
| 400 | quatrocentos | kwatrose~tus |
| 500 | quinhentos | ki~Jetus |
| 600 | seiscentos | sejse~tus |
| 700 | setecentos | setese~tus |
| 800 | oitocentos | ojtose~tus |
| 900 | novecentos | novese~tus |

ordem). Exemplo, o algoritmo de centenas, utiliza o de dezenas, que por sua vez, serve-se do código que trata de unidades. Fica visível que acaba ocorrendo um cascateamento entre os algoritmos, caracterizando reutilização de código. A prática facilita futuras expansões - normalização de números maiores.

Outra responsabilidade do algoritmo principal é fazer a validação da entrada. Impedir que *strings* contendo outros caracteres e espaços em branco, assim como retirar zeros à esquerda. Apesar de simples, estas são etapas essenciais para o bom funcionamento do algoritmo. Atualmente, o sistema compreende números entre 0 (zero) a 9.999.999 (nove milhões novecentos e noventa e nove mil novecentos e noventa e nove).

Mesmo a ordem do sequenciamento dos algoritmos sendo das maiores representações para as menores, será mostrado o inverso, por ser mais didático e simples de entender. Antes de prosseguir, uma breve explicação sobre a notação contida nas Figuras 3.2 até 3.8. A variável ORDEM indica a quantidade de dígitos do algarismo. UNIDADE, DEZENA e CENTENA enumeram os dígitos começando pela direita. Acrescentando que a análise é feita da esquerda para a direita, dígito a dígito, iniciando com o algoritmo correspondente.

A Figura 3.2 exibe o fluxograma do mais básico dos métodos. O processo consiste

unicamente de captar o número analisado e consultar a Tabela 3.6.

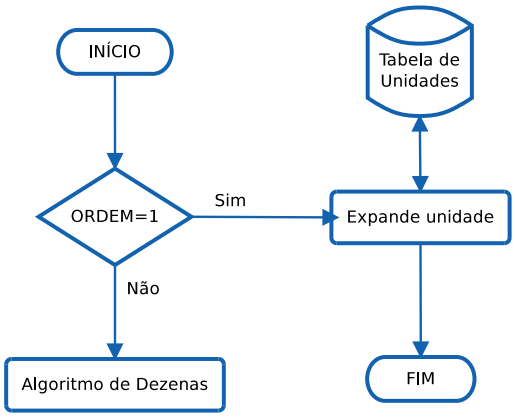


Figura 3.2: Fluxograma do algoritmo para expansão de unidades de números cardinais.

O desenvolvimento de numerais das classes das dezenas já necessita um pouco mais de procedimentos. O motivo é que as dezenas na faixa entre dez e dezenove não seguem as regras comuns à categoria. Consequentemente, primeiro verifica-se o dígito inicial do número (DEZENA), e caso este seja igual a um, consulta-se a Tabela 3.7. Caso contrário, consulta-se a Tabela 3.8 e posteriormente verifica se é necessário expandir a unidade. A sequência de passos é ilustrada na Figura 3.3.

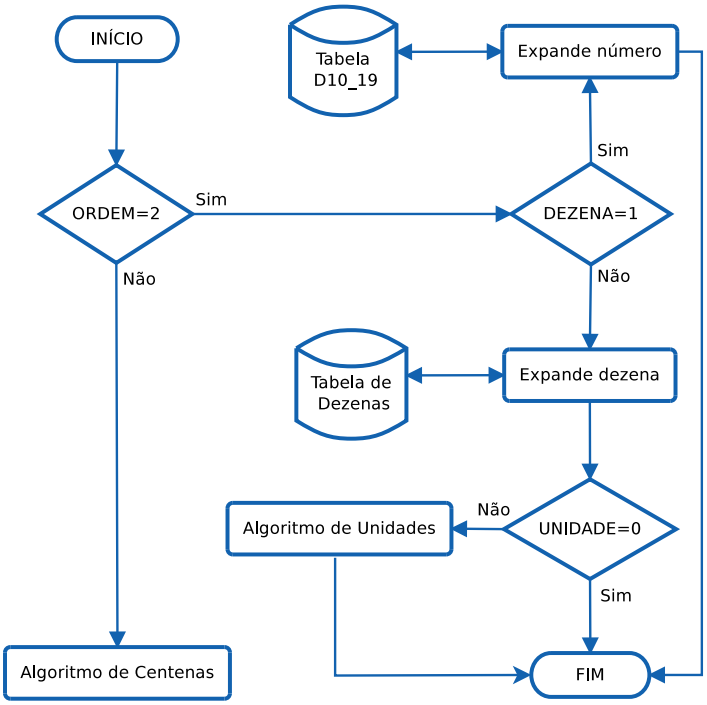


Figura 3.3: Fluxograma do algoritmo para expansão de dezenas de números cardinais.

Para as centenas, a conjunto de passos continua seguindo os mesmos fundamento. Pri-

meiramente expande-se a centena, baseado na Tabela 3.9. Prosseguindo, o algoritmo verifica se aplicará a expansão da unidade e da dezena. Veja na Figura 3.4.

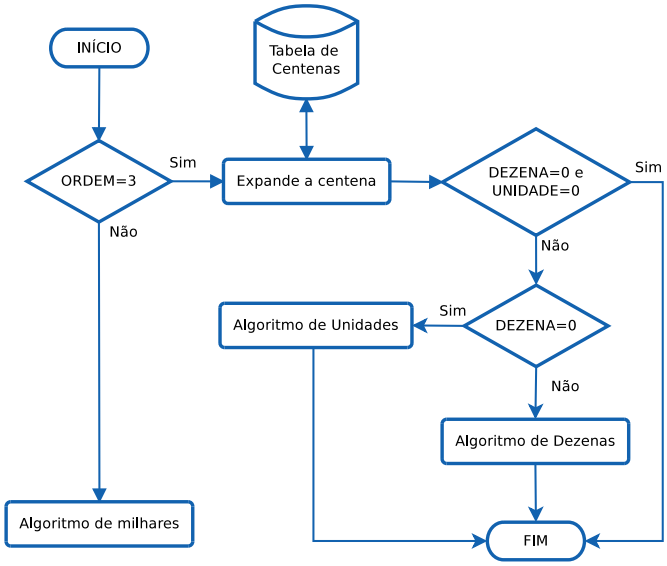


Figura 3.4: Fluxograma do algoritmo para expansão de centenas de números cardinais.

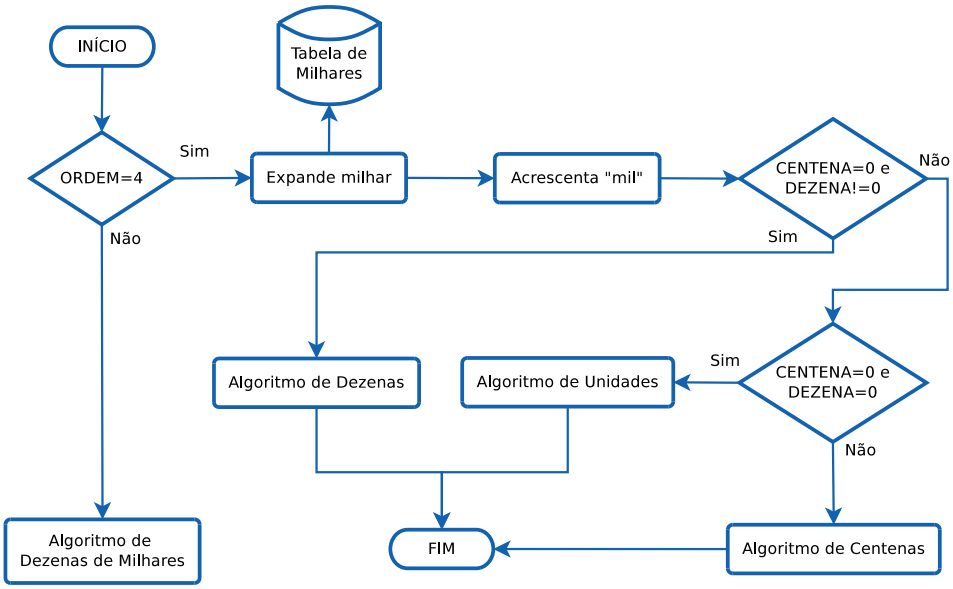


Figura 3.5: Fluxograma do algoritmo para expansão de milhares de números cardinais.

A Figura 3.5 enuncia o autômato que cuida de um problema na expansão de cardinais: a inclusão da conjunção <e >. O problema é pertinente entre os algarismos de milhar e centena, onde hora se faz ausente, hora presente (ex. $1_e001$, $1_e100$ vs 1120, 1540). Via de regra, a conjunção não é inserida, porém quando verificados certos padrões nas casas da dezena e unidade, o algoritmo opta por adicioná-lo. A mini-rotina não é mostrada na Figura 3.5 por questões de espaço.

A partir deste ponto, os algoritmos pode ser considerados cópias dos anteriores com algumas pequenas alterações. Por exemplo, para estender dezenas e centenas de milhar, é utilizado um esquema similar ao algoritmo de milhar, porém aplicando os algoritmos de dezena e centena, respectivamente. As Figuras 3.6 e 3.7 expõem as pequenas diferenças.

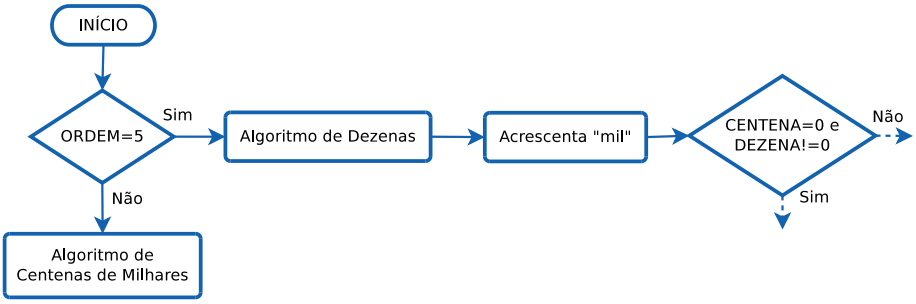


Figura 3.6: Fluxograma do algoritmo para expansão de dezenas de milhares de números cardinais.

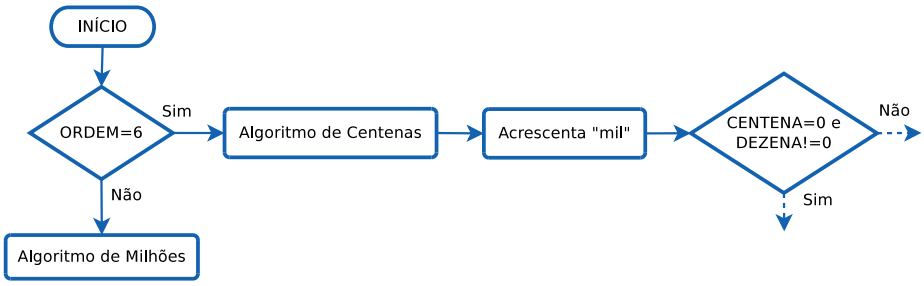


Figura 3.7: Fluxograma do algoritmo para expansão de centena de milhares de números cardinais.

A Figura 3.8 apresenta a expansão das unidades de milhões e reforça as características de reutilização de código da metodologia.

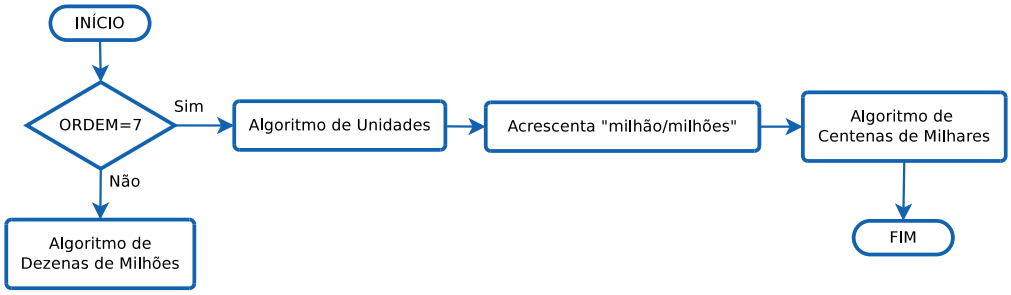


Figura 3.8: Fluxograma do algoritmo para expansão de milhões de números cardinais.

Como visto nas tabelas e figuras anteriores, para que o autômato funcione bem é necessário que todos os algoritmos sejam utilizados em conjunto, respeitando a ordem imposta.

#### 3.2.6.2 Algoritmo de expansão de números ordinais

Os esquemas para normalização de números ordinais é parecida com o apresentado na Seção 3.2.6.1. Primeiramente, a classe se diferencia por flexionarem em gênero e número. Entretanto, a flexão em gênero é a mais frequente e a única tratada aqui. A variação em número foi deixada para ser tratada fora do expansor, pois exige um tratamento que foge do escopo do módulo atual. Além do mais, o plural de números ordinais é caracterizado simplesmente pela inclusão do <s>no final da palavra, o que não dificulta um tratamento externo.

O processo também é baseado em dicionário e recebe o atenuante dos ordinais obedecerem uma formação consolidada a partir das dezenas, tornando as rotinas mais simples. O algoritmo principal funciona de maneira semelhante ao dos cardinais, porém também deve verificar a presença dos expoentes <º>e <ª>, que anunciam numerais ordinais.

A proposta atual transforma somente até 9999(º, ª)(nove milésimo(a) nongentésimo(a) nonagésimo(a) nono(a)), pois valores maiores são difíceis de encontrar. As Tabelas 3.10, 3.11 e 3.12 indicam as listas de extensões.

Tabela 3.10: Lista das extensões das unidades de números ordinais

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 1º, ª | primeiro(a) | pri~mejr(u,a) |
| 2º, ª | segundo(a) | segu~d(u,a) |
| 3º, ª | terceiro(a) | teXsejr(u,a) |
| 4º, ª | quarto(a) | kwaXt(u,a) |
| 5º, ª | quinto(a) | ki~t(u,a) |
| 6º, ª | sexto(a) | sest(u,a) |
| 7º, ª | sétimo(a) | sEtSi~m(u,a) |
| 8º, ª | oitavo(a) | ojtav(u,a) |
| 9º, ª | nono(a) | no~n(u,a) |

O autômato para números ordinais é tão simples quanto o da Seção 3.2.6.1, que seu funcionamento pode ser visto em um único fluxograma (Figura 3.9). Assim como o anterior, futuras ampliações na faixa de valores do algoritmo podem ser naturalmente implementadas.

Tabela 3.11: Lista das extensões das dezenas de números ordinais

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 10º, ª | décimo(a) | dEsi˜m(u,a) |
| 20º, ª | vigésimo(a) | vigEzi˜m(u,a) |
| 30º, ª | trigésimo(a) | trigEzi˜m(u,a) |
| 40º, ª | quadrigésimo(a) | kwadrigEzi˜m(u,a) |
| 50º, ª | quinquagésimo(a) | ki˜kwagEzi˜m(u,a) |
| 60º, ª | sexagésimo(a) | seksagEzi˜m(u,a) |
| 70º, ª | septuagésimo(a) | septuagEzi˜m(u,a) |
| 80º, ª | octuagésimo(a) | oktuagEzi˜m(u,a) |
| 90º, ª | nonagésimo(a) | nonagEzi˜m(u,a) |

Tabela 3.12: Lista das extensões das centenas de números ordinais

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 100º, ª | centésimo(a) | se˜j˜(u,a) |
| 200º, ª | ducentésimo(a) | duse˜tEzi˜m(u,a) |
| 300º, ª | tricentésimo(a) | trese˜tEzi˜m(u,a) |
| 400º, ª | quadrigentésimo(a) | kwadriZe˜tEzi˜m(u,a) |
| 500º, ª | quingentésimo(a) | ki˜Ze˜tEzi˜(u,a) |
| 600º, ª | sexcentésimo(a) | sesse˜tEzi˜m(u,a) |
| 700º, ª | septicentésimo(a) | septSise˜tEzi˜m(u,a) |
| 800º, ª | octigentésimo(a) | oktSiZe˜tEzi˜m(u,a) |
| 900º, ª | nongentésima(o) | no˜Ze˜tEzi˜m(u,a) |

Tabela 3.13: Lista das extensões dos milhares de números ordinais

| Numeral | Expansão | Transcrição fonética |
|---|---|---|
| 1000º, ª | milésimo(a) | milEzi˜m(u,a) |
| 2000º, ª | dois(duas) milésimos(as) | dojs milEzi˜m(u,a) |
| 3000º, ª | três milésimos(as) | trejs milEzi˜m(u,a) |
| 4000º, ª | quatro milésimos(as) | kwatru milEzi˜m(u,a) |
| 5000º, ª | cinco milésimos(as) | si˜ku milEzi˜m(u,a) |
| 6000º, ª | seis milésimos(as) | sejs milEzi˜m(u,a) |
| 7000º, ª | sete miléesimos(as) | setSi milEzi˜m(u,a) |
| 8000º, ª | oito milésimos(as) | ojtu milEzi˜m(u,a) |
| 9000º, ª | nove milésimos(as) | novi milEzi˜m(u,a) |

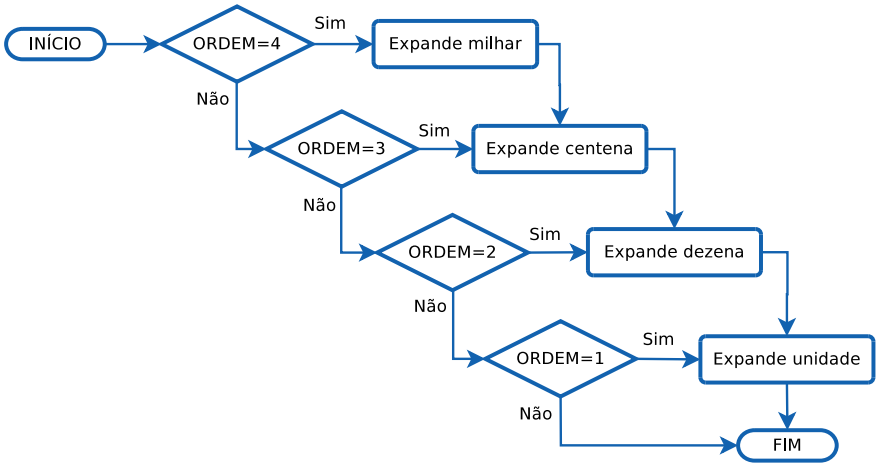


Figura 3.9: Fluxograma do algoritmo expansão de números ordinais

#### 3.2.6.3 Algoritmo de expansão de números romanos

No nível de complexidade de conversão do numeral, o expansor de número romanos é o mais simples dos exibidos neste trabalho, pois a leitura será em cardinal ou ordinal (com base na Tabela 3.14). De certa forma, este conversor utiliza diretamente os algoritmos das subseções anteriores. O primeiro passo consiste em saber como o numeral deve ser lido. Após a definição, aplica-se a transformação necessária.

Tabela 3.14: Lista das conversões de números romanos

| Numeral romano | Conversão em cardinal | Conversão em ordinal |
|---|---|---|
| I | um | primeiro(a) |
| II | dois | segundo(a) |
| III | três | terceiro(a) |
| IV | quatro | quarto(a) |
| V | cinco | quinto(a) |
| VI | seis | sexto(a) |
| VII | sete | sétimo(a) |
| VIII | ... | .. |

O primeiro problema é saber se o que está sendo lido é realmente um algarismo romano. Alguns algarismos podem ser facilmente confundidos com outros padrões na escrita. O algarismo <I>pode representar a leitura de um estrangeirismo, como em “I love you”. Para tratar estes casos, seria necessário incorporar ao processo um leitor de estrangeirismo, o que não foi tratado neste trabalho. Para tratar outras questões, foi criada uma lista contendo as exceções dos numerais restantes e que podem ser confundidos com outros padrões. A lista contempla somente os grafemas <V> e <X>.

- Grafema <V> - Se a palavra anterior for um numero cardinal: 10V ↔ dez volts.

- Grafema <X> - Se o grafema formar algumas das seguintes sequências:
  - cromossomo(a) X
  - Malcolm X
  - triplo X
  - raio(s) X
  - geração X
  - senhor X

Confirmado que o grafema é um algarismo romano, o segundo desafio é saber quando ler em cardinal ou ordinal. Para esta etapa, utilizamos algumas regras que levam em consideração a morfologia das palavras em volta. Caso alguma das duas afirmações abaixo não sejam obedecidas, o numeral é lido como cardinal.

- Romano para ordinal:
  - A palavra anterior é um nome próprio - D. Pedro I ↔ dom pedro primeiro.
  - A palavra posterior começa com letra maiúscula - XI Seminário de Iniciação Científica ↔ décimo primeiro seminário de iniciação científica.
- Romano para cardinal: outros casos.
  - lista I ↔ lista um.
  - século XX ↔ século vinte.

### 3.2.7 Tradutor de casos especiais

Apesar do português brasileiro ser uma língua bastante vasta e rica em detalhes, é possível identificar padrões na escrita de diversas estruturas que não podem ser convertidas diretamente pelos módulos posteriores dentro do *front end* (conversor grafema-fonema, silabador e marcador de vogal tônica). Estes componentes só entendem um número reduzido de caracteres, que se limitam somente ao alfabeto do idioma em questão. Além disso, caracteres como </>e <@>(barra e arroba, respectivamente) são abstrações que são feitas para empregá-los em diversos contexto na escrita da língua portuguesa. Datas, horas, endereços de Internet e *emails* e números telefônicos são padrões que constantemente adotam estas abstrações.

A primeira medida seria traduzir diretamente estes caracteres, aproveitando o conversor de símbolos mostrado na Seção 3.2.4. Apesar de prático, este procedimento pode não gerar

bons resultados, porque o contexto interfere na tradução. Tome-se, como exemplo, o caso do símbolo <->presente na Tabela 3.3. Entre números, este *token* representa a operação matemática de subtração e deve ser lido como "menos". Porém, quando se lê endereços de correio eletrônico, aplicar a interpretação anterior não é o recomendável, sendo mais comum falar "hífen"ou, até mesmo, "traço".

Para cuidar de situações como a descrita no parágrafo anterior, foi criado o tradutor de casos especiais. Sua função é identificar padrões de texto em alto nível e normalizá-los. Não chega a ser uma tarefa de muita complexidade, porém, cada padrão exige um ajuste individualizado. Datas e horas são compostas basicamente de números, mas tem formatação e leitura diferentes. Devido a isto, é necessário uma rotina de datas e outra para horas. O intuito de separar estes dos cuidados dos módulos anteriores, é dotar o sistema com a capacidade de identificar e normalizar diversos modelos que são consolidados na escrita do português.

Para cuidar destas ocasiões especiais, criou-se um módulo específico. Pode ser processado antes mesmo das outras etapas e utiliza constantemente o módulo descrito na Seção 3.2.6, já que muitos dos padrões envolvem numerais.

Tabela 3.15: Lista das extensões das centenas de números ordinais.

| Caso | Exemplo | Normalização |
|---|---|---|
| Email | igorcouto@gmail.com | igorcouto arroba gmail ponto com |
| URL | globo.com | globo ponto com |
| | www.globo.com | w w w ponto globo ponto com |
| | http://www.globo.com | h t t p dois pontos barra barra w ... |
| | https://www.globo.com | h t t p s dois pontos barra barra w ... |
| Telefone | 1111-1111 | um um um um um um um um |
| | (11)11111-1111 | onze um um um um ... |
| | (11) 1111-1111 | onze um um um um ... |
| CPF | 111.111.111-11 | um um um um um um um um um dígito onze |
| IP | 111.111.111.111 | um um um ponto um um um ponto um .... |
| Data | 02[/-.]06[/-.]00 | dois de junho de dois mil |
| | 02[/-.]06[/-.]2000 | dois de junho de dois mil |
| Hora | 23:09 | vinte e três horas e nove minutos |
| | 5:04 | cinco horas e quatro minutos |
| | 03:20:02 | três horas vinte minutos e dois segundos |

O intuito desta etapa é aumentar a eficácia do normalizador, identificando padrões que precisam ser tratados de forma mais específica. A tarefa é realizada com o uso de expressões regulares, que verificam se determinado conjunto de palavras obedece a determinado caso. Cada expressão trata um caso da Tabela 3.15. O *tokens* são verificados caso-a-caso (expressão-a-expressão) e de forma sequencial. Refinar o modelo, seria implementar mais expressões regulares depois ou entre as já existentes, dependendo da situação.

### 3.2.8 Regras de uso

Como em qualquer sistema baseado em regras, a utilização do normalizador apresentado deve ser tomada de certos cuidados. Escrever levianamente e desobedecer as normas gramaticais é partir para uma péssima utilização do normalizador, não permitindo extrair ao máximo do seu desempenho. A lista abaixo pode ser classificada como um manual de uso para as entradas do texto que será normalizado:

1. Escreve da maneira correta. Algumas 'variações' são tratados, porém respeite a norma culta:
   - Ex: Você e não vc. Hoje e não hj.
2. Abreviações são geralmente normalizadas, porém evite inventar ou abreviar levianamente e demasiadamente as palavras:
   - Ex: Hoje fui ao méd. e voltei com péssimas notícias.
3. Se abrir um parêntese, feche-o. Estes deve sempre vir juntos das palavras em que estão cercando e separados do caracter anterior e junto do posterior, caso este seja um sinal de pontuação:
   - Ex: Os profissionais (advogados, médicos, dentistas, engenheiros), quando exercem a profissão por conta própria, são considerados segurados autônomos.
   - Ex: Belo Horizonte (MG) tem uma infra-estrutura.
4. Travessão é diferente de hífen, saiba quando usá-los adequadamente. Basicamente, o uso é:
   - Travessão [—]. No início de frase, indica mudança de locutor, no meio é usado para ressaltar uma expressão ou como sinal de dois pontos[1]:

[1]Veja em http://pt.wikipedia.org/wiki/Travess%C3%A3o

  - Ex: — Bom dia, meu filho.
  - Ex: Pedro — para alegria geral — escreveu 15 linhas somente!
  - Ex: No Congresso tem sempre o mesmo tipo de gente — os corruptos.
- Hífen [-]. Ligar elementos de palavras compostas[2]:
  - Ex: Comprei um semi-novo.
- Obs: Meia-risca é tratada como hífen no normalizador.

5. Quando utilizar o travessão como sinal de pontuação, separe-o tanto do caractere anterior quanto do posterior:
   - Pedro — para alegria geral — escreveu 15 linhas somente!
6. Com exceção dos sinais das indicações 3 e 4, todos os sinais de pontuação são sempre agrupados à palavra anterior e separados da posterior:
   - Ex: Estou com você. E hoje...

## 3.3 Análise fonética

Depois do tratamento inicial dado pelo módulo PLN, o texto de entrada do sistema TTS encontra-se pronto para ser analisado foneticamente. O objetivo dos componentes da análise fonética é converter o texto em uma sequência de fonemas [1]. Outros tipos de informações também são atribuídas, como a divisão das palavras em unidades menores (sílabas) e a subsequente determinação das partes tônicas (sílaba tônica). Estas informações são importantes, porque vão auxiliar o sistema TTS na formação da prosódia - ritmo e entonação correlatos à fala - durante a transformação de texto em fala [1]. Por este motivo, em [27], Holmes também denomina esta etapa como **determinação da prosódia**.

A análise fonética do sistema TTS deste trabalho realiza as tarefas de (1) conversão grafema-fonema, (2) divisão silábica e (3) marcação da sílaba tônica. O primeiro processo consiste na transcrição fonética das palavras, ou seja, converter grafema em fonema (G2P). A divisão silábica particiona a palavra em unidades menores (silabação). E a marcação da sílaba tônica aponta qual das sílabas tem maior entonação (tonicidade).

Na literatura, não são poucos os trabalhos que apresentam soluções para realizar a análise fonética de textos em português [9, 71–75], sendo que a maioria destas publicações apresenta métodos baseados em regras para realizar as análises - o que dá a vantagem de

[2]Veja em http://pt.wikipedia.org/wiki/H%3%ADfen

que o modelo não precisa ser treinado. Para incorporar algum destes métodos ao trabalho, é necessário que sejam apresentados os algoritmos, ou seja, a sequência de passos necessárias para aplicar as regras. Porém, raramente os autores dispõem essas informações durante as publicações. Por exemplo, em [73], o autor apresenta aproximadamente uma centena de regras para realizar a transcrição fonética para o PB, mas não especifica como elas devem ser aplicadas e nem disponibiliza códigos que apliquem tal método.

Além disso, nenhuma das publicações acima apresenta uma solução fechada para as três tarefas (G2P, silabação e tonicidade) necessárias no sistema TTS do trabalho. O objetivo aqui não é desenvolver uma nova abordagem para a análise fonética e, sim, utilizar conjuntos de regras já desenvolvidas e testadas. A estrutura de regra utilizada aqui baseou-se nas regras dos trabalhos [74, 75], pois são recentes e mostraram resultados satisfatórios.

A Figura 3.10 ilustra os componentes integrantes do módulo de análise fonética. O módulo analisa palavra por palavra e as três etapas são feitas na seguinte ordem: silabação, G2P e tonicidade. Na Figura 3.10, é possível verificar que cada passo adiciona informações na saída. As regras de cada uma das etapas não serão apresentadas aqui, por serem bastante extensas, estarem bem explicadas nas publicações de origem e necessitarem de um amplo conhecimento sobre regras fonológicas para o PB. Apesar disso, serão explicados de onde cada etapa foi baseada.

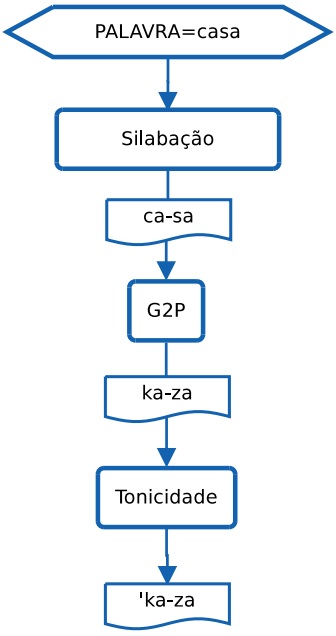


Figura 3.10: Fluxograma da ordem de execução dos componentes da análise fonética para a palavra *casa*.

### 3.3.1 Silabação

O conversor G2P de [74] não realiza a silabação ou marcação da sílaba tônica. Entretanto, estas duas tarefas são necessárias, porque determinarão a entonação da fala. Para implementar o algoritmo, foram utilizados o conjunto de regras descrito em [75]. A ideia principal de Braga [75] é que toda a sílaba tem a vogal como núcleo central, sendo ou não cercada por consoantes ou semivogais. As regras de [75] indicam somente como as letras em torno de uma vogal devem ficar dispostas. Partindo disto, desenvolveu-se um algoritmo que faz uma busca pelas vogais na palavra. Cada vez que uma vogal é localizada, determinado conjunto de regras é aplicado. Segundo os autores, as regras foram testadas em um extrato aleatório de texto da base de dados CETEN-Folha, atingindo uma taxa de erro de 0,71%.

### 3.3.2 Conversão grafema-fonema

O componente G2P baseou-se no trabalho de [74]. Nele, os autores descrevem um conversor grafema-fonema para o PB, baseado em um conjunto de regras descritas em [73]. Uma vantagem dos algoritmos baseados em regras é que, quando comparados com classificadores, como árvores de decisão, o alinhamento léxico não é necessário, desde que o programa não precise ser treinado para gerar as próprias regras. Em outras palavras, a solução proposta, baseada em critérios fonológicos pré-estabelecidos, é suprir o sistema de acordo com a linguagem para a qual a aplicação é direcionada. Sua arquitetura não depende de etapas intermediárias, ou seja, outras rotinas, tais como a divisão silábica ou identificação plural. Há um conjunto de regras para cada grafema e uma ordem específica de aplicação é assumida, de forma sequencial. Em primeiro lugar, as regras mais específicas são consideradas até uma regra caso geral, que termina o processo. Nenhuma análise de coarticulação entre as palavras foi executada, e o conversor G2P [74] trata apenas palavras isoladas. Outro recurso deste conversor é que ele identifica a vogal tônica na palavra (tonicidade). Esta informação acaba sendo aproveitada para determinar a sílaba tônica. Segundo Siravenha [74], o conversor teve taxa de erro de 1% no seu melhor desempenho.

### 3.3.3 Marcação de sílaba tônica

Identificar a sílaba tônica se beneficiou do fato de que o conversor G2P [74], apesar de não separar em sílaba, já ser capaz de identificar a vogal tônica. Construiu-se um algoritmo que armazena a vogal tônica e espera o resultado da silabação. Associando estas duas informações, é possível identificar a sílaba tônica. Desta maneira, o processo de marcar a sílaba tônica pode

ser encarada como a junção entre os dois algoritmos anteriores. Em [75], ainda é apresentado uma rotina para determinação a tonicidade, porém, como já se tinha a rotina de [74] pronta e que se mostrou eficiente, preferiu-se optar pela última.

# Capítulo 4

# Construção do sistema TTS

Após o desenvolvimento dos recursos do módulo de análise do texto e utilizando algumas das ferramentas disponibilizadas pela plataforma MARY, partiu-se para a confecção do sistema TTS para o PB. Este capítulo trata da explicação de como os recursos do Capítulo 3 serão empregados em conjunto com outros também desenvolvidos por este trabalho.

O objetivo deste capítulo é servir como um guia para aqueles que desejam construir seu próprio sistema TTS com auxílio do MARY. Os próprios desenvolvedores deste software disponibilizam alguns tutoriais para a utilização da plataforma [21], entretanto, estes são genéricos, já que seguem a tendência de fazer a plataforma ser utilizada para qualquer língua.

Neste capítulo, serão detalhados os passos que foram feitos para a criação do sistema TTS deste trabalho e será iniciado com um breve explanação de como a plataforma MARY funciona e qual a sua serventia. Depois, serão apresentados algumas etapas para a construção do sistema TTS em si.

## 4.1 A plataforma MARY

A motivação para o uso do MARY neste trabalho é que a plataforma é completamente escrita em Java e suporta tanto a síntese concatenativa como a síntese baseada em HMM. MARY significa ***M**odular **A**rchitecture for **R**esearch on speech s**Y**nthesis* e é um recente *framework* de código aberto para sistemas TTS. Como indica o nome, é concebida para ser altamente modular, com um foco especial sobre a transparência e acessibilidade nas etapas de processamento intermediárias da síntese de voz. Atualmente, a plataforma suporta as línguas alemã, inglesa e tibetana.

A plataforma objetiva ser uma ferramenta flexível para a pesquisa, desenvolvimento e

ensino na área de síntese de voz. A característica modular permite que sejam inseridas novas línguas e criadas novas vozes. Contudo, é importante lembrar que, apesar de já contar com suporte para algumas línguas, o MARY sozinho não constitui um sistema TTS completo. Ele somente possui as ferramentas que são comuns em todos os sistemas TTS e que não são dependentes de nenhuma língua. Por exemplo, os códigos referentes ao treinamento das HMM's são os mesmos para qualquer língua, assim como o algoritmo de extração dos parâmetros MFCC. Já o módulo PLN, integrante da etapa de análise de texto, é dependente de língua, ou seja, é específico da língua em questão. São estes últimos componentes que o MARY não disponibiliza.

Dar suporte a uma nova língua no MARY significa preencher as lacunas (módulos) que são dependentes da língua. E criar uma nova voz é fazer uso deste suporte para o treinamento e ajuste do modelo que será responsável pelo processo de síntese. Nas últimas versões da plataforma, bastante esforço foi feito para a criação de ferramentas de apoio que facilitassem que os usuários inserissem novas línguas. Algumas destas serão apresentadas nas seções posteriores. No decorrer do trabalho, não serão explicadas e nem detalhadas as rotinas internas sobre estas ferramentas, porque tutoriais e documentos podem ser encontrados em [21].

A arquitetura do MARY é do tipo cliente-servidor. Assim, o sistema TTS que será criado é composto em aplicação servidor, que contém os componentes que realizarão a síntese, e a aplicação cliente, que é responsável por interfacear com o usuário e fazer requisições para que o servidor execute alguma tarefa. Um mesmo servidor pode ter suporte para diferentes línguas e ficar a espera de requisições de um ou mais clientes em um porta especificada pelo usuário. Por sua vez, o cliente apresenta para o usuário as configurações suportadas pelo servidor a quem está conectado, como as línguas e os tipos de vozes que podem ser sintetizadas. A Figura 4.1 ilustra a interface de aplicação cliente do MARY sendo executado no navegador de Internet Safari.

Outro ponto a favor da utilização do MARY é que, construir um sistema TTS começando das etapas mais básicas requer muito esforço e tempo. Muito seria gasto implementando rotinas que já são conhecidas no meio acadêmico e que não trariam nada de novo. Como o projeto do MARY também vem sendo alvo de constante atenção por meio da comunidade científica, encontrou-se um modo de dar visibilidade e impulsionar o estudo de síntese de voz para o PB. Um dos objetivos deste trabalho é justamente incentivar o estudo de tecnologias texto-fala para o PB, fortalecendo a pesquisa local.

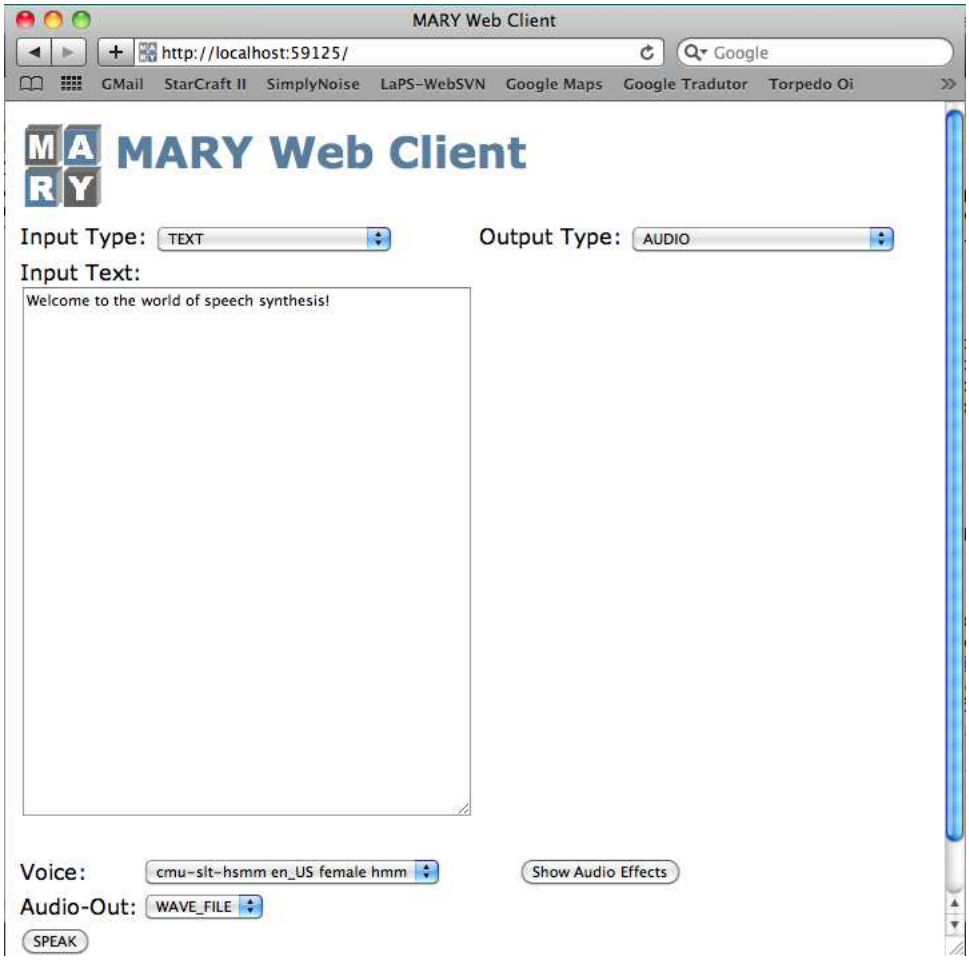


Figura 4.1: Interface da aplicação cliente da plataforma MARY.

## 4.2 Construção do sistema TTS.

Pode-se dividir a construção do sistema TTS no MARY em duas etapas: a criação do suporte para o PB e a construção do modelo da voz. A primeira consiste em fazer uso dos recursos desenvolvidos no Capítulo 3 para fazer com que o MARY seja capaz de tratar requisições para sintetizar sentenças em PB. A segunda diz respeito ao treinamento do modelo com as HMM's e uma base de dados. Pode-se dizer que o primeiro procedimento deve organizar o módulo de análise de texto e o segundo irá gerar o motor de síntese.

### 4.2.1 Suporte para o português brasileiro

Cada língua necessita de um módulo de análise de texto específico, devido às especificidades de cada uma, como conjunto de fonemas, entonação, alfabeto, etc. Portanto, o MARY precisa de um conjunto de arquivos para cada língua que for sintetizar. A ferramenta de apoio

que será utilizada nesta etapa é a *Transcription GUI*. Em [21], um diagrama funcional mostra os requisitos e procedimentos que devem ser feitos para a realização desta etapa. A Figura 4.2 é uma adaptação do diagrama de [21].

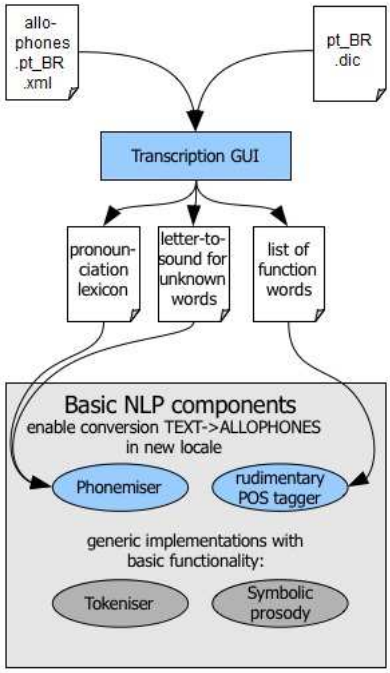


Figura 4.2: Diagrama funcional da construção do suporte para o português brasileiro no MARY. Adaptada de [21]

A Figura 4.2 exibe o procedimento para a elaboração do suporte (conjunto de arquivos) a uma nova língua. Nela, é possível ver o *software* disponibilizado por criar estes arquivos, intitulado de **Transcription GUI**. Sua função é criar os arquivos que dirão ao MARY como tratar requisições específicas de cada idioma (silabação, conversão grafema-fonema, sílaba tônica). Alternativamente, o usuário com mais conhecimento sobre o assunto pode criar sua própria ferramenta com técnicas mais avançadas se assim desejar. A única restrição seria manter a formatação utilizada pelo MARY.

Os dois arquivos no topo do processo são o alfabeto fonético (allo-phones.pt_BR.xml) e a lista de palavras (pt_BR.dic) com as respectivas transcrições. O primeiro não é constituído somente com o alfabeto fonético (SAMPA), pois para cada fone existe um conjunto de atributos que o distingue dos demais, chamados de características distintivas [32]. Na fonética, uma característica distintiva é a unidade básica no fonema, que permite distinguí-lo dos demais. Cada fonema é dotado de especificidades que podem ser traduzidas em vários parâmetros. Se é vogal ou consoantes, vozeado ou não-vozeado, posição da língua na hora da pronúncia, etc. Sendo assim, no dicionário fonético, além do fonema, existem as características distintivas de

cada unidade. O arquivo construído para este trabalho pode ser visualizado no Apêndice C.

Construir um arquivo modelo para o PB com as características distintivas de fone requer conhecimento de fonética. A solução ideal seria contactar (contratar) um foneticista, para que este elaborasse o arquivo da maneira que melhor se adequasse a nossa língua. Entretanto, a metodologia adotada foi a seguinte: usar o arquivo para o inglês que já estava disponível em [21]. Apesar da estrutura ser a mesma, todos os fonemas são diferentes, assim como as suas características distintivas. Para parametrizar as vogais, recorreu-se à tabela de vogais do IPA (*Internetional Phonetic Alphabet*), que é a notação fonética padrão para todas as línguas. A método foi ouvir todas as vogais presentes no IPA e fazer o pareamento com as do português brasileiro. Quando se descobria o símbolo correspondente ao som, procurava-se no gráfico os outros atributos. Ainda que o atual recursos possa não ser o ideal, este pareceu não impactar negativamente na qualidade da voz através de testes de escutas informais.

A segunda entrada para o *Transcription GUI* (pt_BR.dic) é uma lista com transcrições fonéticas de algumas palavras. Nem todas as palavras da lista precisam estar transcritas, porém, é necessário que pelo menos algumas estejam. Com as entradas que estiverem transcritas, a ferramenta irá, por meio de treinamento com árvores de regressão, gerar um conversor grafema-fonema na forma de um transdutor finito de estado [76]. Um pequeno exemplo da lista de entrada pode ser visto na Tabela 4.1

Tabela 4.1: Tabela com as transcrições que devem ser passados ao Transcription Gui

| Palavra | Transcrição |
|---|---|
| abacate | a-ba-'ka-tSi |
| abriu | a-'briw |
| barraca | ba-'Ra-ka |
| casa | 'ka-za |

Transdutores finitos de estado são máquinas de estados finitas que permitem produzir saídas gravando-se a estrutura de entradas previamente apresentadas [76]. Nesse caso, as entradas são as transcrições passadas pelo usuário. Logicamente, a fidelidade do transdutor ao modelo sobre o esquema adotado para a conversão grafema-fonema será diretamente relacionada com a qualidade e quantidade das transcrições usadas. Poucos exemplos podem fazer com que a máquina de estados não crie estruturas que não realizem a transcrição de maneira satisfatória. A Figura 4.3 expõe a interface do programa.

Além do conversor grafema-fonema, o *Transcription GUI* irá gerar outros arquivos. Na nomenclatura do MARY, estes arquivos são chamados de componentes básicos para o

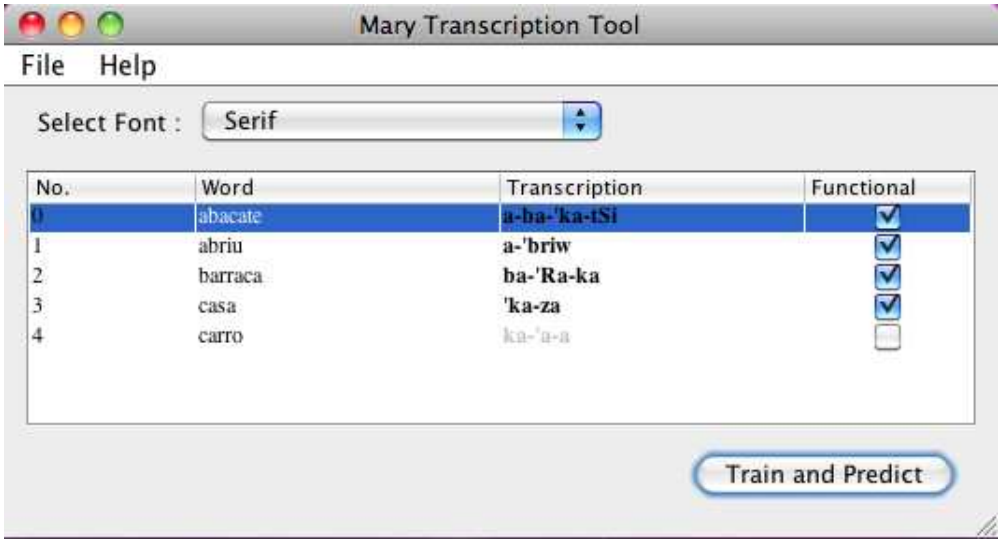


Figura 4.3: Interface do Transcription GUI.

módulo PLN, porque estes pelo menos devem ser capazes de fazer a conversão grafema-fonema (fonetizar) e realizar um etiquetador rudimentar. Este etiquetador somente indica se a palavra foi ou não transcrita anteriormente. A lista de arquivos para o módulo PNL:

- pt_BR_lexicon.fst - transdutor finito de estado que fará a conversão grafema-fonema das palavras.
- pt_BR_pos.fst - transdutor finito de estados responsável pelo etiquetamento rudimentar.
- pt_BR_lexicon.dict - lista com as palavras passadas pelo usuário.

A arquitetura modular do MARY permite que se utilize outra ferramenta em detrimento do *Transcription GUI*, caso o usuário desejar. Pesquisadores mais experientes poderiam criar etiquetadores mais sofisticados e somente respeitar a formatação especificada.

Após a criação destes arquivos, é necessário copiá-los para locais específicos dentro dos diretórios de instalação do MARY. Estes caminhos devem ser também especificados dentro do arquivo de configuração do módulo, chamado de pt_ BR.config. Este último será lido quando o MARY é iniciado, habilitando o PB.

### 4.2.2 Construção do modelo de voz

A segunda etapa é ajustar e treinar o modelo com as HMM's que serão responsáveis pela síntese. Contudo, antes de começar as etapas de treino, alguns procedimentos são realizados. Em linhas mais gerais, pode-se dividir a construção do modelo de voz em duas partes: ajustar

a motor de síntese e criar a voz. A primeira parte significa saber quais e quantos são os fones a serem utilizados, quais modelos de HMM farão a síntese, quais arquivos de áudio serão usados no treino, etc. A segunda pode ser resumida como a execução do algoritmos e rotinas de treinamento da HMM's. Ainda que a documentação também contemple outras formas de síntese providas pelo MARY, será concentrada a atenção unicamente nos procedimentos que dizem respeito a preparação de um motor de síntese baseado em HMM.

Assim como na parte de preparação do suporte para um novo idioma, este estágio também recebeu uma ferramenta para auxiliar os usuários, chamada de *VoiceImport*. Como os procedimentos aqui são bem mais complexos para serem executados no terminal de uma computador, o *VoiceImport* se mostra bastante útil, pois ele se torna uma espécie de interface mais amigável para a manipulação de programas típicos de síntese de voz e HMM's (HTS), que nem sempre - ou quase nunca - são simples de entender. Contudo, algum conhecimento sobre o assunto é requisito mínimo para utilizá-la. A Figura 4.4 exibe a interface do *VoiceImport*.

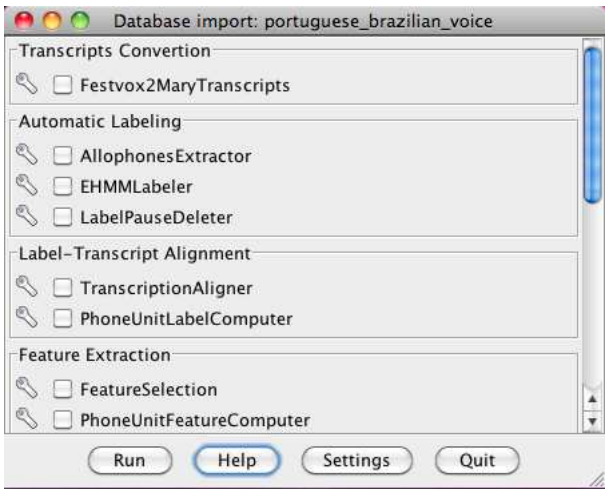


Figura 4.4: Interface da ferramenta *VoiceImport*.

Sendo mais específico, o *VoiceImport* trata-se de um conjunto de componentes que fazem o interfaceamento com algum ferramenta que o usuário precise usar no processo de criação de uma voz. A maior vantagem é o encapsulamento de vários comandos, que em outras circunstâncias o usuário deveria passar diretamente pela linha de comando do sistema operacional. Por exemplo, o módulo Festvox2MaryTranscripts transforma as transcrições do Festival [77] para o formato do próprio MARY. Já em HMMVoiceDataPreparation, o componente verifica se todos os programas necessários (hts,sox,etc) estão devidamente instalados, e caso contrário, instala-os. Pode parecer uma mera ferramenta guia, porém, como será visto adiante, estas facilidades tiram do usuário a preocupação com a formatação e tratamento dos dados, fazendo com que ele se concentre nas questões mais diretamente ligados à construção

de um TTS.

Como o MARY suporta tanto a síntese baseada em HMM's quanto a concatenativa, o *VoiceImport* foi preparado para ter rotinas que cuidem das duas técnicas. Devido a esta característica, muitos dos componentes do *VoiceImport* não serão mencionados. Os interessados em fazer um TTS usando o modelo concatenativo são convidados a fazê-lo, inclusive podendo utilizar os recursos já disponibilizados em capítulos anteriores. Porém, por estar fora do escopo deste trabalho, não encontrarão mais detalhes aqui. Para esta jornada, é aconselhável que se leia [21].

Os componentes exigidos para a tarefa podem ser classificados em dois grupos: as subtarefas de preparação dos dados e de treinamento das HMM's. Por questões de espaço, é inviável e improdutivo se aprofundar demasiadamente - vasculhar código - em cada componente. Ater-se-á às mais importantes técnicas empregadas em cada um deles. Para maiores detalhes, todos estes componentes estão disponíveis e documentados em [21]

### 4.2.3 Preparação dos dados

Verificação dos programas instalados, dos caminhos dos arquivos de texto e áudio, geração dos rótulos fonéticos, definição de que características serão utilizadas no modelo são algumas das rotinas implementadas por este componente.

#### 4.2.3.1 Módulo Festvox2MaryTranscriptions

Unicamente converte as transcrições textuais do projeto Festival para o formato que será aproveitado durante as etapas posteriores. No Festival, as transcrições de toda a base são armazenadas em um único arquivo. No MARY, as transcrições são separadas por sentença, criando um *arquivo_xx.txt* correspondente para todo *arquivo_xx.wav*

#### 4.2.3.2 Módulo HMMVoiceDataPreparation

Verifica se os programas externos estão instalados (sox, sptk, hts, perl, etc). Caso contrário, instala-os. Essa etapa se faz necessária, porque a plataforma não implementa em código as muitas das rotinas necessárias para as etapas de treinamento e síntese. Vários componentes fazem chamadas a estes programas para realizar diversas tarefas, por exemplo: Praat para determinação do *pitch*, EHMM para etiquetamento do rótulos fonéticos, HTS/HTK para o treinamento dos modelos HMM's.

#### 4.2.3.3 Módulo AllophonesExtractor

Na Seção 4.2.1, mostrou-se que é necessário suprir o servidor de recursos para tratar as requisições de um determinado idioma. Apesar de quase todo o suporte só ser utilizado quando o TTS estiver em funcionamento, é possível ver que alguns componentes precisam realizar tarefas típicas do módulo de processamento do texto. Em AllophonesExtractor, extraem-se os fones dos arquivos de texto a partir do suporte habilitado no servidor. Esse procedimento é necessário, porque o usuário só entra com o áudio e texto.

A conversão grafema-fonema é feita com o transdutor finito de estados. Esta é uma etapa importante, porque caso o modelo não seja bom, as etapas posteriores serão diretamente afetadas, assim como a qualidade da voz criada. Imagine que o conversor não realize sua tarefa de forma adequada, transformando erradamente. Na palavra *carro*, se o resultado for *kaaa* ao invés de *kaRu*, (1) será impossível o rotulamento fonético - próximo componente - fazer o alinhamento forçado e (2) o HTS/HTK receberá trechos não correspondentes para treinar as HMM's.

#### 4.2.3.4 Módulo EHMMLabeller

Segmentar e rotular os fonemas de forma precisa são tarefas importantes tanto para a síntese baseada em HMM quando a concatenativa, pois esta informação é utilizada para classificar e selecionar apropriadamente as unidades fonéticas, que mais tarde serão utilizadas no treinamento e estimação dos modelos. Segundo [78], assegurar a eficiência neste processo é essencial para produzir uma voz de qualidade. O autor também evidencia que o desalinhamento das fronteiras dos fonemas, troca entre o fonema rotulado e o pronunciado e até mesmo a presença de um ruído de fundo são problemas recorrentes.

Os métodos que possuem maior exatidão são as técnicas de rotulação manual feitas por especialistas em linguística. Entretanto, este tipo de abordagem consome muito tempo, fazendo com que diversos outros métodos automáticos fossem criadas nos últimos anos [79] [80] [81]. Ainda que estejam sujeitas a erros, de modo que muitas vezes são seguidos por uma fase de verificação (correção) manual, estes mostram-se muito mais úteis e práticos em trabalhos onde a base é muito grande ou não existam especialistas para a tarefa.

Devido ao apresentado no parágrafo anterior, a ferramenta de construção de voz disponibiliza o Sphinx [82] por meio dos componentes *SphinixLabelingPreparator* e *SphinxLabeler* e o EHMM [77] através do *EHMMLabeler*. Este último foi o utilizado no trabalho, por ser mais bem ajustado à construção de vozes sintéticas. Nele, modelos contínuos com uma Gaussiana por estado, *left-to-rigth* e independentes de contexto com 13 coeficientes mel-cepstrais

da escala mel são usados para se obter os limites entre fonemas, numa técnica conhecida como alinhamento forçado. Dependendo do tamanho da base de áudio, o processo pode demorar várias horas.

#### 4.2.3.5 Módulo LabelPauseDeleter

Componente meramente técnico, sendo usado somente para evitar problemas de formatação nos componentes subsequentes da ferramenta de construção de voz.

#### 4.2.3.6 Módulo TranscriptionAligner

Cria arquivos em formato XML, que contém o texto e informações como: transcrição, divisão silábica dos fones, sílaba tônica, duração. Tais arquivos serão utilizadas em etapas posteriores.

#### 4.2.3.7 Módulo PhoneUnitLabelComputer

Converte os arquivos gerados pelo EHMM (Subseção 4.2.3.4) para o formato do MARY.

#### 4.2.3.8 Módulo FeatureSelection

Variabilidade é uma palavra chave em processamento de fala e que pode assumir valores antagônicos. Se em reconhecimento de voz, a variabilidade de um fonte pode levar a diversos problemas, na síntese, a falta da variabilidade em um modelo pode levar a monotonia e pouca naturalidade da voz sintetizadas [83]. A voz sintetizada de um sistema TTS não pode soar natural sem um tratamento das variabilidades da duração dos sons e pausa na fala, bem como o posicionamento destas. Modelar as características e a estrutura temporal da fala é um interesse particular da tecnologia de síntese, representando o limiar entre os aspectos cognitivos e mecânicos da voz.

Em uma visão global, as características que derivam do trato vocal e dos mecanismos articulatórios são iguais em todas os idiomas. Porém, existem aspectos específicos, que são conectados com fatores típicos da língua e falante [84]. Segundo [85], é muito grande a probabilidade de que o mesmo texto falado por duas pessoas tenham características temporais e prosódicas diferentes.

O componente *FeatureSelection* oferece as opções de características que podem ser utilizadas na construção da voz.

#### 4.2.3.9 Módulo PhoneUnitFeatureComputer

Depois de escolhidos quais aspectos serão empregados no modelo, o componente *PhoneUnitFeatureComputer* computa vetores de características para as sequências de fonemas de cada transcrição através do servidor MARY. Sendo mais específico, ocorre a aplicação do processo de determinação da sílaba tônica e a busca das características distintivas de cada fonema, recursos estes que foram anteriormente habilitados como suporte na Seção 4.2.1. Portanto, é requisito que o servidor MARY esteja rodando.

Os aspectos mencionados são em sua maioria de representações fonéticas (altura da vogal, tipo de consoante, etc) e contextuais (fonema anterior, próximo fonema) [86]. Tais vetores são úteis no treinamento das HMM's de duração, nos modelos das árvores de classificação e regressão e nas HMM's geradoras do parâmetros acústicos.

Para os leitores que ainda não conseguiram imaginar a serventia deste recurso ao final do processo de síntese, basta lembrar que no Brasil, existem diferentes variações do português brasileiro, reproduzindo diversas formas de execução da língua portuguesa. Estas variações, fazem com que o ritmo e entonação da fala sejam também diferentes. Esta etapa tem também (não esquecer das desigualdades existentes entre falantes) a responsabilidade de captar estas feições e repassá-las ao produto final - a voz sintetizada.

#### 4.2.3.10 Módulo PhoneLabelFeatureAligner

Os rótulos e vetores com as características dos fones precisam estar alinhados. Em *PhoneLabelFeatureAligner*, é feita a comparação entre os dois elementos. Caso eles não estejam alinhados, existe um problema. Nesse caso, a ferramenta deixa o usuário decidir como resolver o empasse. Ou editando o rótulo, ou alterando o vetor.

### 4.2.4 Treinamento das HMM's

Para a criação de vozes baseadas em HMM, são utilizados os *scripts* de treinamento disponíveis nos códigos de demonstração (*demos*) do HTS [13], que foram adaptados para o *VoiceImport*. Estes arquivos vêm de forma embarcada, ou seja, dentro da ferramenta. As alterações são poucas e não chegam a modificar substancialmente o processo. Uma delas é a adição das magnitudes em algumas bandas de frequência do sinal de voz no dados de treino. Dessa forma, os códigos tiveram que ser configurados para tratar este novo parâmetro acústico.

#### 4.2.4.1 Módulo HMMVoiceConfigure

A etapa de treino começa com a verificação dos arquivos de áudio e as transcrições fonéticas, pois devem estar nos formatos apropriados. Alguns parâmetros podem ser definidos nesta parte. Na Tabela 4.2, alguns deles são exibidos.

Tabela 4.2: Tabela com parâmetros que podem ser definidos no *HMMVoiceConfigure*. Adaptado de [21].

| Parâmetro | Valor |
|---|---|
| HMMVoiceConfigure.dataSet = | brazilian_portuguese_voice |
| HMMVoiceConfigure.speaker = | igor_couto |
| HMMVoiceConfigure.lowerF0 = | 40 (male=40, female=80) |
| HMMVoiceConfigure.upperF0 = | 280 (male=280, female=350) |

#### 4.2.4.2 Módulo HMMFeatureSelection

Escolha definitiva de quais características serão utilizadas para a captura da prosódia. Além disso é alternativa a inserção de outras *features*, a parte das explicadas na Subseção 4.2.3.8, e que serão úteis no treinamento das HMM's.

#### 4.2.4.3 Módulo HMMVoiceMakeData

Como dito no Capítulo 2, em síntese de voz, as cadeias de Markov não são treinadas com os sinais de áudio *in natura*. Ao invés disso, uma série de parâmetros (coeficientes cepstrais, frequência fundamental, variâncias globais, etc.) são extraídos da fala e, estes sim, são modelados.

#### 4.2.4.4 Módulo HMMVoiceMakeVoice

No Capítulo 2, são mostrados os procedimentos referentes a técnica HTS na etapa de treino. É neste componente que será feito o treinamento das HMM's referentes a cada fonema do alfabeto fonético.

#### 4.2.4.5 Módulo HMMVoiceInstaller

Este último componente tem somente a tarefa de realizar a instalação dos arquivos que foram gerados pelos componentes anteriores. No momento que o MARY for realizar o processo de síntese, os procedimentos que ele utiliza serão iguais aos descritos no Capítulo 2 na parte de síntese. Um módulo PLN fará a análise do texto, as HMM's serão selecionadas e concatenadas, gerando, posteriormente, a voz sintetizada.

# Capítulo 5

# Considerações Finais

Métricas e formas de avaliação são ferramentas importantes para avaliar a evolução de qualquer sistema. É como diz o ditado: "Não se pode melhorar o que não se pode avaliar". Neste capítulo, as seções iniciais dissertam sobre alguns aspectos de testes de sistemas TTS e a metodologia de avaliação que foi aplicada no sistema deste trabalho, assim como as suas características. Posteriormente, os resultados são analisados, encerrando-se o trabalho com tópicos importantes sobre as futuras melhorias.

## 5.1 Teste do sistema

A qualidade do sinal de fala sintetizado é um atributo fundamental de um sistema TTS [1]. Muitos codificadores de áudio (celular, telefones, rádio, etc) podem ser avaliados a partir de diversos fatores, tais como largura de banda do sinal, taxa de *bits*, imunidade ao ruído, atraso e erro. Todos estes são formas objetivas de se avaliar a qualidade de um sistema. Entretanto, para sistemas TTS é necessário uma outra forma de avaliação, que ao contrário dos fatores anteriormente ditos, não podem ser quantificados em um simples número ou meramente fixados a partir de uma fórmula matemática. Outro ponto importante diz respeito a um dos principais objetivos de uma sistema TTS - ser **inteligível** e **natural**. O primeiro aspecto significa compreender a informação que está sendo transmitida, enquanto o segundo mede o quanto este sistema está próximo da voz humana.

Para sistemas TTS, a medida de qualidade de voz mais amplamente utilizada é o *Mean Opinion Score* (MOS) [3], que é o resultado da média de pontuações atribuídas por um conjunto de indivíduos não treinados [1]. O MOS pode ser visto com um protocolo de avaliação de qualidade subjetiva recomendado pela *International Telecommunication Union*

(ITU) [87] para equipamentos e sistemas de telecomunicação. O ITU apresenta dois grupos de teste: os testes de opinião de conversação e os testes de opinião de escuta. Cada um destes grupos tem vários métodos de avaliação, sendo que neste trabalho será aplicado um método de opinião de escuta.

A metodologia de avaliação deste trabalho é uma adaptação do método *Absolute Category Rating* (ACR) [88]. Diz-se adaptação, porque nem todos os requisitos e aspectos do ACR foram reproduzidos durante o teste por serem impossíveis de aplicar neste primeiro momento. O ACR especifica alguns padrões que são difíceis de controlar durante a avaliação do som, como: tamanho, nível de ruído máximo e tempo de reverberação do ambiente, forma de gravação do áudio, calibração do sinal, etc. Das muitas destas especificações, foram seguidas as relativas às amostras de áudio, escalas de pontuação, instruções aos indivíduos do teste e análise e relatório dos resultados.

As amostras de áudio foram geradas pelo sistema a partir de sentenças encontradas aleatoriamente em anúncios de jornais, revistas e endereços na Internet. O ACR especifica que estas amostras devem ter entre 3 à 5 segundos, não sendo muito curtas e nem demasiadamente longas. Durante o teste, os entrevistados não tem conhecimento das sentenças que lhe serão passadas. As sentenças utilizadas para o teste de opinião de escuta deste trabalho - ao todo 20 sentenças - podem ser vistas na Tabela 5.1.

Tabela 5.1: Sentenças utilizada no teste de opinião de escuta

| | |
|---|---|
| o carro é azul | você vai ter que ficar calmo |
| a noite é sempre misteriosa | nada precisa ser feito |
| congresso vota ficha limpa | o tribunal cancelou a liminar |
| maria precisa de dinheiro | ação de terroristas matam cinco |
| venda de ingressos começa amanhã | promoção entregará presentes |
| embratel chega com banda larga em belém | roma decepciona |
| cansaço preocupa nadal | projeto leva educação ao interior |
| vendaval deixa rastro de destruição | campanha incentiva prevenção |
| confira os destaques do barra pesada | conferência começa hoje |
| há um defeito no sistema | é necessário manter o controle |

A escala de pontuação obedece o padrão exposto na Tabela 5.2. Esta escala é utilizada para quantificar de forma subjetiva os aspectos de inteligibilidade e naturalidade. A cada amostra de áudio ouvida, duas notas devem ser atribuídas pelo entrevistado, sendo uma para cada aspecto. Segundo Taylor [3], é possível que inteligibilidade e naturalidade sejam

analisadas por métodos distintos. Entretanto, aplicou-se o ACR sobre os dois, com a intenção de avaliar o desempenho do sistema.

Tabela 5.2: Escala de pontuação para o teste de opinião de escuta.

| Qualidade do áudio | Pontuação |
|---|---|
| Excelente | 5 |
| Bom | 4 |
| Razoável | 3 |
| Ruim | 2 |
| Pobre | 1 |

Além do ACR, também foi utilizada a *Word Error Rate* (WER) [89]. Nela, é calculada a porcentagem de palavras erradas que o entrevistado entendeu em cada sentença. A WER é utilizada para avaliar a inteligibilidade do sistema de forma objetiva e é definida de acordo com a Equação 5.1.

$$WER = \frac{D - R - A}{W} * 100 \tag{5.1}$$

onde $D$ é a total de palavras que não foram reconhecidas, $R$ é o total de palavras que foram entendidas com outra palavra e $A$ é o total de palavras que não faziam parte da sentença, mas foram adicionadas pelo entrevistado. $W$ representa a quantidade total de palavras na frase. Relacionada a WER, ainda existe a *Word Accuracy Rate* (WAR), que define a porcentagem de palavras acertadas em cada sentença. A WAR pode ser facilmente calculada, computando-se $100 - WER$.

Antes de começar cada teste, é apresentado ao indivíduo a escala da Tabela 5.2. Além disso, o indivíduo é instruído a fazer a avaliação sobre dois ângulos diferentes e independentes para cada amostra de áudio apresentada:

- É possível entender o que está sendo dito? A mensagem está clara? Está difícil compreender? (inteligibilidade)
- A voz é natural? É produzida por um ser humano? É artificial? (naturalidade)

Quanto a quantidade de indivíduos entrevistado, a documentação do ITU não especifica um número exato. Para esta avaliação, foram entrevistados 10 candidatos, onde cada entrevista teve duração média de 20 à 25 minutos. Os entrevistados eram, em sua maioria, alunos de graduação e não possuíam nenhum vínculo com o trabalho ou pesquisa na área de síntese de voz.

## 5.2 Outros sistemas

Além do sistema TTS deste trabalho, outros sistemas foram procurados de forma a se obter uma análise comparativa. A escolha destes outros sistemas tornou-se um problema, dado a dificuldade de se encontrar outros *softwares* de código aberto que realizem síntese para o PB. Os principais sistemas operacionais do mercado (Windows, Linux e MacOSX) possuem sintetizadores de voz, porém quase todos direcionados para o inglês e outras poucas línguas.

Apesar da dificuldade, dois sintetizadores foram encontrados. Um comercial, de onde se busca obter uma avaliação do estado da arte em sistemas TTS e outro que foi desenvolvido pela comunidade acadêmica e é bastante utilizado em conjunto com tecnologias assistivas. Por questões de clareza, é bom dizer que os testes de opinião de escuta foram aplicados de maneira idêntica em todos os sistemas participantes.

### 5.2.1 Raquel da Nuance

A Nuance[1] é uma empresa multinacional que trabalha com tecnologias relacionadas à voz. Possui e desenvolve vários produtos que tem como finalidade fazer uso da voz para executar comandos em computadores e equipamentos eletrônicos, aumentando a interação homem-máquina. Um dos grupos de produtos são os sintetizadores de voz em 40 idiomas, disponibilizados em mais de 50 vozes diferentes.

Para o PB, a Nuance vende o sintetizador denominado de Raquel, que reproduz a voz de mulher jovem. A empresa não torna acessível muitas informações de como seus produtos são desenvolvidos. Entretanto, em listas de discussão sobre síntese de voz, é dito que é a síntese concatenativa é a técnica empregada nestes produtos.

A utilização da Raquel nesta avaliação comparativa é justificada pela sua altíssima qualidade. Em testes de escutas informais, não se encontrou outra voz em PB que possuísse qualidade maior que esta. Porém, é importante frisar que a Loquendo[2] também desenvolve vozes para o PB com qualidade similar.

[1] http://www.nuance.com/

[2] http://www.loquendo.com/

### 5.2.2 LianeTTS

O LianeTTS [3] é um sistema que nasceu da cooperação entre o Serviço Federal de Processamento de Dados (SERPRO) e o Núcleo de Computação Eletrônica (NCE) da Universidade Federal do Rio de Janeiro (UFRJ). Foi lançado em abril de 2010 e é atualmente utilizado em telecentros como parte de uma solução para acesso à computadores por pessoas com deficiência.

O programa é um *front end* para o MBROLA [18]. O projeto do MBROLA objetiva criar um conjunto de sintetizadores de fala para quantas línguas forem possíveis, sem aplicação comercial e fins comerciais. De iniciativa da *Faculté Polytechnique de Mons*, na Bélgica, o sintetizador do MBROLA trabalha com a síntese concatenativa através de difones e já disponibiliza diversos idiomas em seu endereço na Internet.

## 5.3 Resultados

De maneira geral, o sistema TTS deste trabalho obteve desempenho superior ao LianeTTS, ficando atrás da voz Raquel da Nuance. Nas subseções posteriores serão apresentados os resultados obtidos nesta avaliação. Primeiro por uma visão subjetiva, obtida pelo método ACR, e depois de forma objetiva, analisando a WER.

### 5.3.1 Avaliação subjetiva - ACR

O ACR permitiu que fossem medidos os aspectos da inteligibilidade e da naturalidade. A Figura 5.1 ilustra as médias das pontuações relativas à inteligibilidade. O sistema comercial da Nuance foi o que obteve o melhor desempenho, aproximando-se do nível de qualidade excelente, aproximando-se da pontuação 5. O sistema TTS deste trabalho, caractetizado como TTS na figura, ficou à frente do LianeTTS com uma pontuação considerada acima do nível razoável, acima de 3.

Já as pontuações referentes à naturalidade foram, no geral, mais baixas que as da inteligibilidade. Deve-se ainda ao fato de que mesmo os sistemas TTS mais bem elaborados do mercado ainda não conseguiram reproduzir uma voz que seja facilmente confundida com voz humana. Apesar disso, o sintetizador da Nuance ainda obteve uma alta pontuação, porém se aproximando mais do nível de qualidade bom do que excelente. A naturalidade do sistema

[3] Notícia sobre o *software* no portal do SERPRO: http://www.serpro.gov.br/noticiasSERPRO/2010/abril/serpro-e-ufrj-desenvolvem-leitor-de-telas-para-deficientes-visuais/

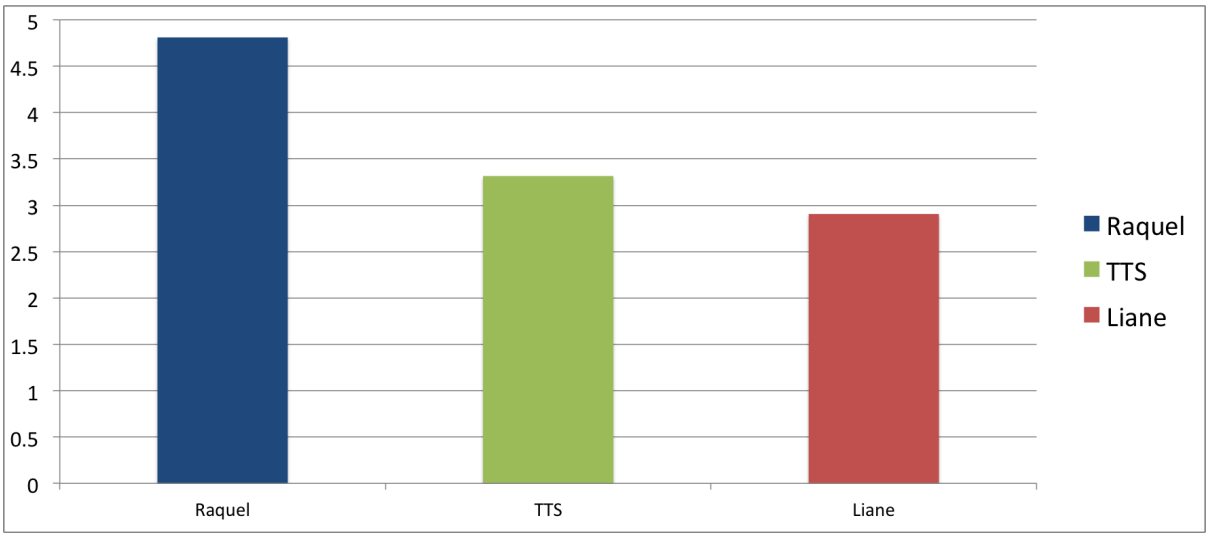


Figura 5.1: Média aritmética simples da inteligibilidade dos sistemas TTS.

deste trabalho também caiu, ficando um pouco abaixo da nível considerado razoável, porém, ainda a frente do *software* distribuído pelo SERPRO.

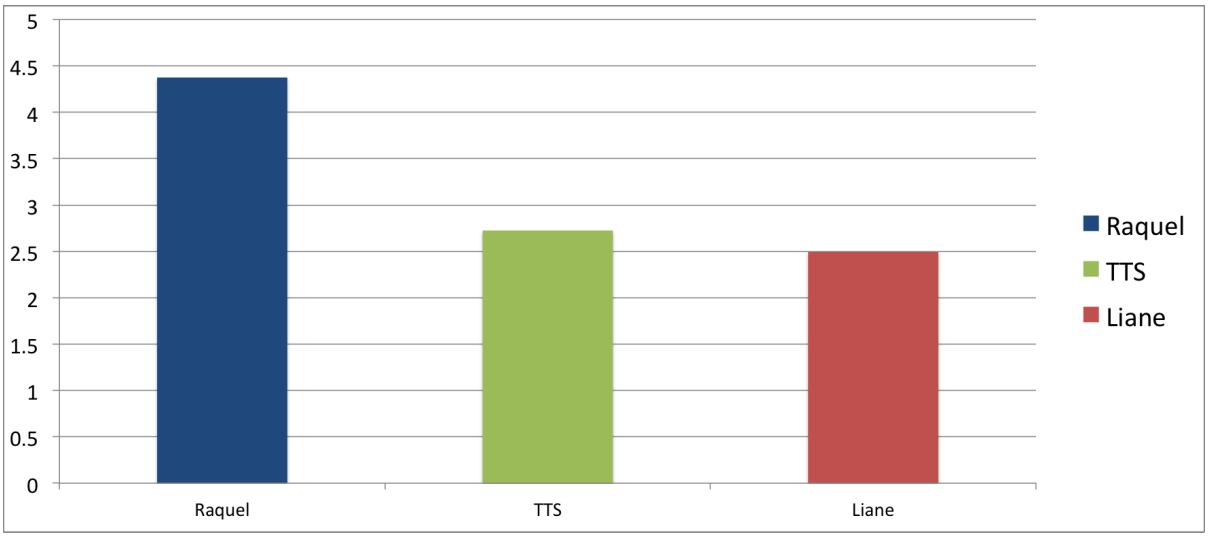


Figura 5.2: Média aritmética simples da naturalidade dos sistemas TTS.

Além das médias aritméticas simples, foram calculadas as modas das pontuações de cada um dos sistemas. Por contabilizar o valor que detém o maior número de observações em uma distribuição, a moda é menos suscetível do que a média aritmética aos efeitos dos *outliers*. Em estatística,*outliers* são observações que se distanciam numericamente do resto do conjunto de dados. Essas observações podem fazer com que a avaliação sobre o desempenho do sistema não seja interpretada da maneira correta. A Tabela 5.3 apresenta o resultado do cálculo da moda sobre as pontuações desta avaliação.

Tabela 5.3: Resultado do cálculo da moda

| Sistema | Inteligibilidade | Naturalidade |
|---|---|---|
| Raquel | 5 | 5 |
| TTS | 3 | 3 |
| LianeTTS | 3 | 3 |

No cálculo da média, o sistema LianeTTS ficou um pouco atrás em termos de pontuação. Entretanto, comparando os sistemas de acordo com a perspectiva da moda, verifica-se que ocorreu um empate entre o TTS deste trabalho e o LianeTTS.

### 5.3.2 Avaliação objetiva - WER

Na avaliação objetiva, o ranking se manteve inalterado. Na Figura 5.3, é visto que o sistema deste trabalho conseguiu a segunda colocação, fazendo com que quase 90% das palavras fossem entendidas pelos entrevistados - WER em torno de 10%. Já o sintetizador LianeTTS obteve o dobro de erros, ultrapassando a marca dos 20%. De acordo com a Figura 5.3, ainda é possível notar o ótimo desempenho do sintetizador Raquel, sendo que a porcentagem de palavras erradas se aproxima de zero.

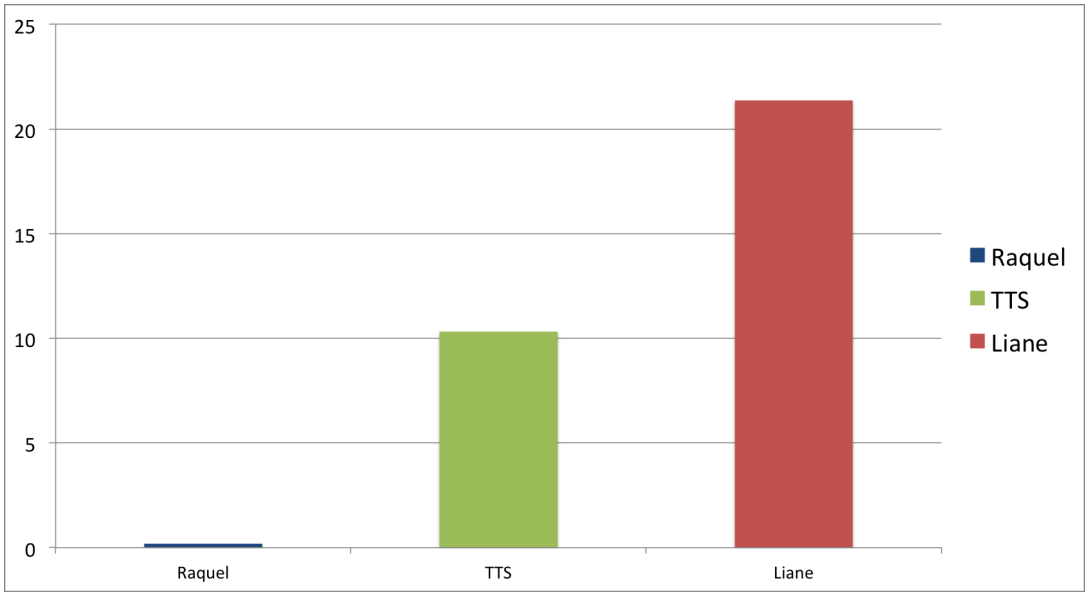


Figura 5.3: Porcentagem de palavras erradas - WER

## 5.4 Trabalhos futuros

A qualidade da voz obtida pela síntese baseada em HMM's ainda pode ser melhor do que a apresentada aqui. O desempenho inferior, neste momento, ainda é resultado principalmente da baixa qualidade e quantidade de arquivos de áudio utilizados como base de dados na etapa de treino do modelo - 18 minutos. Em trabalhos futuros, espera-se conseguir conjuntos de áudio maiores e melhores, objetivando alcançar resultados mais próximos dos melhores sintetizadores comerciais para o PB.

A parte de implementação do sistema também ainda não chegou ao fim. Melhorias e novas técnicas que possam a ser empregadas junto com a síntese HTS podem ser adicionadas ao modelo. Em uma lista não-exaustiva, pode-se enumerar alguns pontos que devem receber atenção:

- Refinar alguns dos módulos PLN, como: aumentar o dicionário do módulo de expansão de abreviações ou inserir novas regras no leitor de siglas.
- Adquirir uma base de áudio com qualidade de estúdio para a etapa de treino do modelo.
- Fazer com que o conversor grafema-fonema, silabador e marcador de sílaba tônica sejam integrados diretamente dentro do MARY.
- Implementar a desambiguação de homófonos e heterofonos.

O sistema ainda está atrelado a plataforma MARY, fazendo com que seja necessário tê-la instalada na máquina em que for feita a síntese. Apesar de bem escrito, o código do MARY carrega rotinas que são desnecessárias à síntese de voz para o PB.

Mesmo que diversos desafios e questões ainda tenham que ser superados, o conhecimento adquirido, o artigo publicado em um congresso internacional, a comunicação e troca de informações entre o grupo FalaBrasil com outras equipes de pesquisa em síntese de voz durante a elaboração deste trabalho, fizeram com que a pesquisa em síntese de voz para o PB na UFPA se torna-se visível perante a comunidade e ganhasse o impulso inicial para desenvolver novas pesquisas.

# Referências Bibliográficas


[1] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing.* Prentice-Hall, 2001.

[2] T. Dutoit, *An Introduction to Text-To-Speech Synthesis.* Kluwer, 1997.

[3] P. Taylor, *Text-to-Speech Synthesis.* Cambridge University Press, 2009.

[4] B. Skinner, *Science And Human Behavior.* Free Press, 1965.

[5] "http://www.laps.ufpa.br/falabrasil," Visited in May, 2010.

[6] L. De C.T. Gomes, E. Nagle, and J. Chiquito, "Text-to-speech conversion system for Brazilian Portuguese using a formant-based synthesis technique," in *SBT/IEEE International Telecommunications Symposium*, 1998, pp. 219–224.

[7] J. Solewicz, A. Alcaim, and J. Moraes, "Text-to-speech system for Brazilian Portuguese using a reduced set of synthesis units," in *ISSIPNN*, 1994, pp. 579–582.

[8] F. Egashira and F. Violaro, "Conversor Texto-Fala para a Língua Portuguesa," in *13th Simpósio Brasileiro de Telecomunicações*, 1995, pp. 71–76.

[9] E. Albano and P. Aquino, "Linguistic criteria for building and recording units for concatenative speech synthesis in Brazilian Portuguese," in *in Proceedings EuroSpeech, Rhodes, Grecia*, 1997, pp. 725–728.

[10] P. Barbosa, F. Violaro, E. Albano, F. Simes, P. Aquino, S. Madureira, and E. Franozo, "Aiuruete: a high-quality concatenative text-to-speech system for Brazilian Portuguese with demisyllabic analysis-based units and hierarchical model of rhythm production," in *Proceedings of the Eurospeech'99, Budapest, Hungary*, 1999, pp. 2059–2062.

[11] I. Seara, M. Nicodem, R. Seara, and R. S. Junior, "Classificação Sintagmática Focalizando a Síntese de Fala: Regras para o Português Brasileiro," in *SBrT*, 2007, pp. 1–6.

[12] R. Maia, H. Zen, K. Tokuda, T. Kitamura, and F. Resende, "An HMM-based Brazilian Portuguese speech synthetiser and its characteristics," *Journal of Communication and Information Systems, v. 21, p. 58-71*, 2006.

[13] "http://hts.ics.nitech.ac.jp/," Visited in May, 2010.

[14] "http://www.nuance.com/realspeak/languages/," Visited in May, 2010.

[15] "http://www.loquendo.com/en/demos/demo_tts.htm," Visited in May, 2010.

[16] "http://www.acapela-group.com/portuguese-brazil-46-text-to-voice.html," Visitado em Maio, 2010.

[17] D. Braga, P. Silva, M. Ribeiro, M. S. Dias, F. Campillo, and C. García-Mateo, "Hélia, Heloísa and Helena: new HTS systems in European Portuguese,Brazilian Portuguese and Galician," in *PROPOR 2010 - International Conference on Computational Processing of the Portuguese Language*, 2010, pp. 27–30.

[18] "http://tcts.fpms.ac.be/synthesis," Visited in May, 2010.

[19] "http://intervox.nce.ufrj.br/dosvox/," Visited in May, 2010.

[20] "http://cslu.cse.ogi.edu/toolkit/," Visited in May, 2010.

[21] "http://mary.opendfki.de/," Visited in June, 2010.

[22] S. Hertz, "Integration of rule-based formant synthesis and waveform concatenation: a hybrid approach to text-to-speech synthesis," sep. 2002, pp. 87 – 90.

[23] C.-H. Lee, S.-K. Jung, and H.-G. Kang, "Applying a speaker-dependent speech compression technique to concatenative tts synthesizers," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 632 –640, feb. 2007.

[24] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating articulatory features into hmm-based parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 1171 –1185, aug. 2009.

[25] P. C. Neto and U. Infante, *Gramática da Língua Portuguesa*. Editora Scipione, 2004.

[26] F. de Saussure, *Curso de Linguística Geral*. Cultrix, 2006.

[27] J. N. Holmes and W. J. Holmes, *Speech Synthesis and Recognition*. T & F STM, 2001.

[28] N. D'Alessandro, "An Introduction to Text-to-Speech Synthesis." [Online]. Available: so-on.be/SO-ON/workshops/voice/pdf/TTS.pdf

[29] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–86, Feb. 1989.

[30] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptative algorithm for mel-cepstral analysis of speech," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1992.

[31] D. Callou and Y. Leite, *Iniciação à Fonética e à Fonologia.* Jorge Zahae Editor, 1995.

[32] R. Jakobson, C. G. M. Fant, and M. Halle, *Preliminaries to speech analysis: the distinctive features and their correlates.* Cambridge : Acoustics Laboratory, Massachusetts Institute of Technology, 1952.

[33] "http://www.ait.pt/recursos/dic_term_ling/index2.htm," Visitado em outubro de 2010.

[34] C. Cunha and L. Cintra, *Nova Gramatica do Portugues Brasileiro.* Edicoes Joao Sa Costa, 2000.

[35] "http://www.radames.manosso.nom.br/gramatica/grafema.htm," Visitado em outubro de 2010.

[36] "http://www.ergative.net/lc/lc_docs/files/alfabeto_fonetico.pdf," Visitado em outubro de 2010.

[37] "International Phonetic Alphabet," Visited in August, 2010. [Online]. Available: http://www.langsci.ucl.ac.uk/ipa/

[38] "http://www.phon.ucl.ac.uk/home/sampa/home.htm," Visited in May, 2010.

[39] J. E. Cahn and J. E. Cahn, "Generating expression in synthesized speech," MIT, Tech. Rep.

[40] I. R. Murray, "Simulating Emotion in Synthetic Speech," Ph.D. dissertation, University of Dundee, UK, 1989.

[41] I. R. Murray and J. L. Arnott, "Implementation and testing of a system for producing emotion-by-rule in synthetic speech," *Speech Communication, 16, 369-390*, 1995.

[42] "http://www.disc2.dk/tools/sgsurvey.html," Visited in May, 2010.

[43] J. van Santen, J. Hirschberg, J. Olive, and R. Sproat, Eds., *Progress in Speech Synthesis.* New York: Springer-Verlag, 1996.

[44] M. Schröder and S. Breuer, "XML Representation Languages as a Way of Interconnecting TTS Modules," in *in Proc. ICSLP, Jeju, Korea*, 2004.

[45] "http://mary.dfki.de/documentation/maryxml," Visitado em setembro de 2010.

[46] "http://nlp.stanford.edu/software/tagger.shtml," Visitado em outubro de 2010.

[47] D. Klatt, "Software for a cascade / parallel formant synthesizer," *Journal of the Acoustical Society of America*, vol. 67, pp. 971–95, 1980.

[48] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis using HMMs with dynamic features," in *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*. Washington, DC, USA: IEEE Computer Society, 1996, pp. 389–392.

[49] K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, and S. Imai, "An Algorithm For Speech Parameter Generation From Continuous Mixture HMMs With Dynamic Features," 1995.

[50] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, pp. 660–663, 1995.

[51] T. Kobayashi, K. Tokuda, T. Masuko, T. Yoshimura, T. Kitamura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," 1999.

[52] K. Tokuda, T. Kobayashi, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[53] K. Tokuda, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Transactions on Information and Systems, vol. E85D, no. 3, pages 455-464*, 2006.

[54] "http://htk.eng.ac.uk," Visitado em outubro de 2010.

[55] M. Barros, R. Maia, K. Tokuda, D. Freitas, and F. R. Jr., "HMM-based European Portuguese speech synthesis," in *Proc. of Interspeech*, 2005, pp. 2581–2584.

[56] S.-J. Kim, J.-J. Kim, and M. Hahn, "Implementation and evaluation of an hmm-based korean speech synthesis system," *IEICE - Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 1116–1119, 2006.

[57] Y. Qian, F. K. Soong, Y. Chen, and M. Chu, "An HMM-Based Mandarin Chinese Text-To-Speech System," in *ISCSLP*, 2006, pp. 223–232.

[58] X. Gonzalvo, I. Iriondo, J. C. Socoró, F. Alías, and C. Monzo, "HMM-based Spanish speech synthesis using CBR as F0 estimator," in *NOLISP*, 2007.

[59] S. Karabetsos, P. Tsiakoulis, A. Chalamandaris, and S. Raptis, "Hmm-based speech synthesis for the greek language," in *TSD '08: Proceedings of the 11th international conference on Text, Speech and Dialogue*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 349–356.

[60] T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Proc. of Eurospeech*, 1997, pp. 2523–2526.

[61] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. of ICSLP*, 2002, pp. 1269–1272.

[62] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi, "Speaking style adaptation using context clustering decision tree for hmm-based speech synthesis," vol. 1, may. 2004, pp. I – 5–8 vol.1.

[63] R. Maia, H. Zen, K. Tokuda, T. Kitamura, and F. G. Resende, "Towards the development of a Brazilian Portuguese text-to-speech system based on HMM," in *in Proc. of the European Conf. on Speech Communication and Technology*, 2003.

[64] H. Zen and T. Toda, "An overview of nitech hmm-based speech synthesis system," in *IEICE Trans. Inf. and Syst*, 2005, pp. 93–96.

[65] C. L. Bennett, "Large Scale Evaluation of Corpus-based Synthesizers: Results and Lessons from the Blizzard Challenge 2005," 2005.

[66] H. kawahara, I. Musuda-Katsuse, and A. de Cheveigné, "Reconstructuring speech representation using a pitch-adaptative time-frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds," *Speech Communication, vol. 27, no. 3-4, pages 187-207*, Apr. 1999.

[67] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-markov model-based speech synthesis system," *IEICE - Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[68] T. Toda and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis," in *Proc. of the European Conf. on Speech Communication and Technology (EUROSPEECH)*, 2005.

[69] J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni, *From text to speech: The MITalk system*. Cambridge University Press, 1987.

[70] D. F. M. B. M. da Silva, "Algoritmos de Processamento da Linguagem Natural para Sistemas de Conversão Texto-Fala em Português," Master's thesis, Facultade de Filoloxía da Universidade da Coruña - Departamento de Galego-Portugués, Francés e Lingüística, May 2008.

[71] I. Trancoso, M. Viana, and F. Silva, "On the pronunciation of common lexica and proper names in European Portuguese," in *2nd Onomastica Res. Colloq*, 1994.

[72] N. J. Mamede, J. Baptista, I. Trancoso, and M. das Graas Volpe Nunes, "Computational processing of the portuguese language, 6th international workshop, propor 2003, faro, portugal, june 26-27, 2003. proceedings," in *PROPOR*. Springer, 2003, pp. 23–30.

[73] D. C. Silva, A. de Lima, R. Maia, D. Braga, J. F. de Morais, J. A. de Morais, and F. G. V. R. Jr., "A rule-based grapheme-phone converter and stress determination for Brazilian Portuguese natural language processing," in *Proc. of IEEE Int. Telecomm. Symposium (ITS)*, 2006.

[74] A. Siravenha, N. Neto, V. Macedo, and A. Klautau, "Uso de Regras Fonológicas com Determinação de Vogal Tônica para Conversão Grafema-Fone em Português Brasileiro," *7th International Information and Telecommunication Technologies Symposium*, 2008.

[75] D. C. Silva, D. Braga, and F. G. V. R. Jr, "Separação das Sílabas e Determinação da Tonicidade no Português Brasileiro," *XXVI Simpósio Brasileiro de Telecomunicações (SBrT'08)*, 2008.

[76] A. Kornai, Ed., *Extended Finite State Models of Language*. Cambridge University Press, 1999.

[77] "http://www.cstr.ed.ac.uk/projects/festival/," Visited in April, 2010.

[78] S. Pammi, M. Charfuelan, and M. Schröder, "Quality Control of Automatic Labelling Using HMM-based Synthesis."

[79] F. Malfrère, O. Deroo, T. Dutoit, and C. Ris, "Phonetic alignment: speech synthesis-based vs. viterbi-based," *Speech Commun.*, vol. 40, no. 4, pp. 503–515, 2003.

[80] J. Kuo, H. Lo, and H. Wang, "Improved hmm/svm methods for automatic phoneme segmentation," in *in Proceedings of Interspeech 2007, Antwerp, Belgium*, 2007.

[81] K. Prahallad, A. W. Black, and R. Mosur, "Sub-phonetic modeling for capturing pronunciation variation in conversational speech synthesis," in *in Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 2006.

[82] "http://www.speech.cs.cmu.edu/sphinx/."

[83] M. Tatham and K. Morton, *Developments in Speech Synthesis*. Wiley, 2005.

[84] M. Mihkla, "Modelling speech temporal structure for estonian text-to-speech synthesis: Feature selection."

[85] N. Campbell, *Prosody, theory and experiment*. 281?334: Kluwer Academic Publishers., 2000, ch. Timing in Speech: a multi-level process.

[86] S. C. Pammi, M. Charfuela, M. Schroder, and O. Turk, "Voice building tool for the mary tts platform." [Online]. Available: http://janus.cs.utwente.nl:8000/twiki/pub/Semaine/Publications/MaryVoiceBuildingTools.pdf

[87] "http://www.itu.int," Visitado em novembro de 2010.

[88] "http://www.itu.int/rec/t-rec-p.800-199608-i/en," Visitado em novembro de 2010.

[89] M. Turunen and C. M. Turunen, "Speech application design and development," 2004.

# Apêndice A

# Rótulos contextuais

Formato do rótulo dependente de contexto. A primeira parte da Tabela A apresenta um exemplo e a segunda explica cada um dos itens. "/S$i$:", "/W$i$:", "/P$i$:" significam a $i$-ésima sílaba, palavra e frase na sentença "/U:". A tabela e o exemplo foram traduzidos de [12]

$m1\hat{}m2 - m3 + m4 = m5/\mathrm{M2} : m6\_m7$
/S1:$s1$_@$s2 - s3$_@$s4 + s5$_@$s6$/S2:$s7\_s8$/S3:$s9\_s10$/S4:$s11\_s12$/S5:$s13\_s14$/S6:$s15$
/W1:$w1$_#$w2 - w3$_#$w4 + w5$_#$w6$/W2:$w7\_w8$/W3:$w9\_w10$/W4:$w11\_w12$
/P1:$p1$_!$p2 - p3$_!$p4 + p5$_!$p6$/P2:$p7\_p8$
/U:$u1$_\$$u2$_&$u3$

| Nível de fonema | |
|---|---|
| m1 | fonema anterior ao fonema anterior |
| m2 | fonema anterior |
| m3 | fonema atual |
| m4 | fonema posterior |
| m5 | fonema posterior ao fonema posterior |
| m6 | posição do fonema atual na sílaba (da esquerda para a direita) |
| m7 | posição do fonema atual na sílaba (da direita para a esquerda) |
| **Nível de sílaba** | |
| s1 | a sílaba anterior é tônica ou não (0 → não; 1 → sim |
| s2 | número de fonemas da sílaba anterior |
| s3 | a sílaba atual é tônica ou não (0 → não; 1 → sim) |
| s4 | número de fonemas da sílaba atual |
| s5 | a sílaba posterior é tônica ou não (0 → não; 1 → sim) |
| s6 | número de fonemas da sílaba posterior |
| s7 | posição da sílaba atual na palavra atual (da esquerda para a direita) |
| s8 | posição da sílaba atual na palavra atual (da direita para a esquerda) |
| s9 | posição da sílaba atual na frase atual (da esquerda para a direita) |
| s10 | posição da sílaba atual na frase atual (da direita para a esquerda) |
| s11 | número de sílabas tônicas antes da sílaba atual na frase atual |

| | |
|---|---|
| s12 | número de sílabas tônicas depois da sílaba atual na frase atua |
| s13 | número de sílabas contando a partir da sílaba tônica anterior para a sílaba atual na sentença |
| s14 | número de sílabas contando a partir da sílaba tônica atual para sílaba posterior na sentença |
| s15 | vogal da sílaba atual |
| Nível de palavra | |
| w1 | etiqueta gramatical da palavra anterior |
| w2 | número de sílabas da palavra anterior |
| w3 | etiqueta gramatical da palavra atual |
| w4 | número de sílabas da palavra atual |
| w5 | etiqueta gramatical da palavra posterior |
| w6 | número de sílabas da palavra posterior |
| w7 | posição da palavra atual na frase atual (da esquerda para a direita) |
| w8 | posição da palavra atual na frase atual (da direita para a esquerda) |
| w9 | número de palavras conteúdo antes da palavra atual na frase atual |
| w10 | número de palavras conteúdo depois da palavra atual na frase atual |
| w11 | número de palavras contando a partir da palavra conteúdo anterior para a palavra atual na sentença |
| w12 | número de palavras contando a partir da palavra conteúdo atual para palavra posterior na sentença |
| Nível de frase | |
| p1 | número de sílabas na frase anterior |
| p2 | número de palavras na frase anterior |
| p3 | número de sílabas na frase atual |
| p4 | número de palavras na frase atual |
| p5 | número de sílabas na frase posterior |
| p6 | número de palavras na frase posterior |
| p7 | posição da frase atual na sentença atual (da esquerda para a direita) |
| p8 | posição da frase atual na sentença atual (da direita para a esquerda) |
| Nível de sentença | |
| u1 | número de sílabas na sentença |
| u2 | número de palavras na sentença |
| u3 | número de frases na sentença |

# Apêndice B

# Árvores de decisão dependentes de contexto.

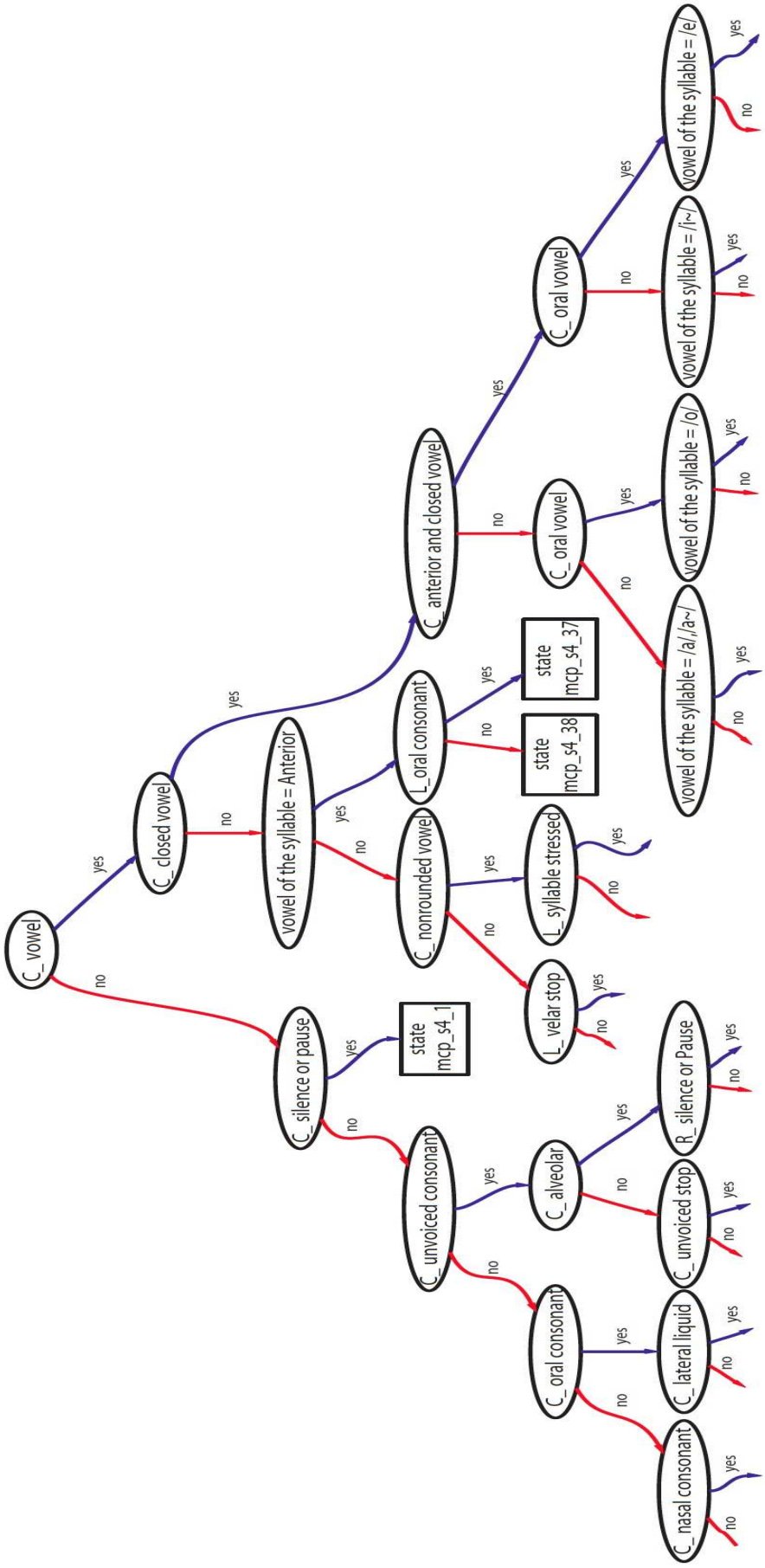


Figura B.1: Exemplo de uma árvore de decisão. Retirada de [12]

# Apêndice C

# Dicionário fonético com as características distintivas.

```
<allophones name="sampa" xml:lang="pt-BR" features="vlng vheight vfront vrnd ctype cplace
    cvox">

 <silence ph="_"/>
 <!-- Oral vowels -->
 <vowel ph="a" vlng="s" vheight="3" vfront="1" vrnd="-"/>
 <vowel ph="E" vlng="s" vheight="2" vfront="1" vrnd="-"/>
 <vowel ph="e" vlng="s" vheight="2" vfront="1" vrnd="-"/>
 <vowel ph="i" vlng="s" vheight="1" vfront="1" vrnd="-"/>
 <vowel ph="O" vlng="s" vheight="2" vfront="3" vrnd="+"/>
 <vowel ph="o" vlng="s" vheight="2" vfront="3" vrnd="+"/>
 <vowel ph="u" vlng="s" vheight="1" vfront="3" vrnd="+"/>
 <!-- Nasal vowels -->
 <vowel ph="a~" vlng="l" vheight="3" vfront="2" vrnd="-"/>
 <vowel ph="e~" vlng="l" vheight="2" vfront="2" vrnd="-"/>
 <vowel ph="i~" vlng="l" vheight="1" vfront="2" vrnd="-"/>
 <vowel ph="o~" vlng="l" vheight="2" vfront="3" vrnd="+"/>
 <vowel ph="u~" vlng="l" vheight="1" vfront="3" vrnd="-"/>
 <!-- Semi-vowels -->
 <vowel ph="w" vlng="d" vheight="2" vfront="2" vrnd="0" ctype="v"/>
 <vowel ph="j" vlng="d" vheight="2" vfront="2" vrnd="0" ctype="p"/>
 <vowel ph="w~" vlng="d" vheight="2" vfront="2" vrnd="0" ctype="v"/>
 <vowel ph="j~" vlng="d" vheight="2" vfront="2" vrnd="0" ctype="p"/>
 <!-- Unvoiced fricatives -->
 <consonant ph="f" ctype="f" cplace="b" cvox="-"/>
 <consonant ph="s" ctype="f" cplace="a" cvox="-"/>
 <consonant ph="S" ctype="f" cplace="p" cvox="-"/>
 <!-- Voiced fricatives -->
 <consonant ph="z" ctype="f" cplace="a" cvox="+"/>
```

```
<consonant ph="v" ctype="f" cplace="b" cvox="+"/>
<consonant ph="Z" ctype="f" cplace="p" cvox="+"/>
<!-- Affricatives -->
<consonant ph="tS" ctype="a" cplace="p" cvox="-"/>
<consonant ph="dZ" ctype="a" cplace="p" cvox="+"/>
<!-- Plosives -->
<consonant ph="b" ctype="s" cplace="l" cvox="+"/>
<consonant ph="d" ctype="s" cplace="l" cvox="+"/>
<consonant ph="t" ctype="s" cplace="d" cvox="-"/>
<consonant ph="k" ctype="s" cplace="v" cvox="-"/>
<consonant ph="g" ctype="s" cplace="v" cvox="+"/>
<consonant ph="p" ctype="s" cplace="l" cvox="-"/>
<!-- Liquids -->
<consonant ph="l" ctype="l" cplace="a" cvox="+"/>
<consonant ph="L" ctype="l" cplace="p" cvox="+"/>
<consonant ph="R" ctype="l" cplace="a" cvox="+"/>
<consonant ph="X" ctype="l" cplace="a" cvox="+"/>
<consonant ph="r" ctype="l" cplace="a" cvox="+"/>
<!-- Nasal consoants -->
<consonant ph="m" ctype="n" cplace="l" cvox="+"/>
<consonant ph="n" ctype="n" cplace="a" cvox="+"/>
<consonant ph="J" ctype="n" cplace="p" cvox="+"/>
</allophones>
```