v"ErbuS

Paulo Jorge Cardoso Carrasqueira

Sistema Online de Conjugação e Pronunciação de Verbos em Português

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores

Setembro de 2015

UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Eletrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Uma Dissertação
Para a Graduação de Estudos em
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Sistema Online de Conjugação e Pronunciação de Verbos em Português

Paulo Jorge Cardoso Carrasqueira

Orientador: Professor Doutor Fernando Santos Perdigão - El Capitan

Júri
Presidente: Professor Doutor Paulo José Monteiro Peixoto
Orientador: Professor Doutor Fernando Santos Perdigão
Vogal: Professora Doutora Lúcia Maria dos Reis Albuquerque Martins
Vogal: Doutora Sara Maria Fernandes Rato e Costa Marques Candeias

Setembro de 2015

Agradecimentos

É com a maior alegria que profiro os meus sinceros agradecimentos, prestando a devida homenagem e atribuindo todo o mérito a quem tornou este trabalho possível:

À minha Mãe, Pai, Irmão, Avós e restante família, pelo apoio, confiança e carinho prestados ao longo deste percurso.

À Ana, por todo o apoio, sorrisos e momentos partilhados, tornado estes 5 anos inesquecíveis.

Ao meu orientador, Professor Doutor Fernando Perdigão, pelo constante apoio, disponibilidade e conhecimentos partilhados que tornaram possível este projeto.

A todos os meus amigos, que desde o primeiro dia estiveram presentes alegrando o meu percurso académico e vida. A vocês devo muito do meu crescimento enquanto pessoa. Levo as nossas amizades para a vida.

À cidade de Coimbra, cidade do conhecimento, das amizades, por todos os momentos que me proporcionou.

"Uma vez Coimbra, para sempre saudade ".

Resumo

O processo de aprendizagem de conjugação e pronunciação de verbos portugueses, requer um enorme esforço por parte dos falantes devido a todas as regras e irregularidades existentes. Este esforço torna-se mais acentuado, para os falantes que utilizam o português como segunda língua. Esta dissertação tem como objetivo proporcionar uma ferramenta para ajudar na aprendizagem da pronunciação e conjugação de verbos portugueses. Essa ferramenta trata-se de um *website* onde é possível conjugar um verbo, nas formas ortográfica e fonética (pronunciação), bem como reproduzir através de síntese de fala, quaisquer flexões verbais tendo em conta a sua transcrição fonética.

Para a criação do sintetizador de fala utilizou-se a ferramenta HMM-based Speech Synthesis System (HTS), que cria os modelos Hidden Markov Models (HMM) utilizados, posteriormente, na síntese de fala. Para criar os modelos HMM, o HTS necessita de efetuar um treino a partir de uma base de dados composta por ficheiros áudio. Estes ficheiros foram gravados em estúdio, tendo sido recolhida uma base de dados com dois locutores, um masculino e outro feminino.

No final do treino são criados os ficheiros que definem a fala sintetizada sendo posteriormente utilizados no sintetizador de fala presente no *website*. O *website* permite ao utilizador obter a conjugação ortográfica e fonética de qualquer verbo e ouvir a sua pronunciação. A conjugação e pronunciação de verbos é realizada recorrendo a duas aplicações *C++* que são evocadas pelo servidor onde se encontra o *website*.

Se o verbo a pronunciar é um dos proferidos pelos locutores da base de dados, então em vez da síntese de fala a partir das flexões verbais, são utilizadas as locuções originais. O *website* permite conjugar e pronunciar qualquer verbo, independentemente deste pertencer, ou não, ao léxico.

Palavras Chave

HMM, HTS, Síntese, Verbos, *Website*

Abstract

The pronunciation and conjugation learning process of Portuguese verbs requires a huge effort from non-Portuguese speakers, since there are several rules and irregularities in the Portuguese grammar. Thus, this dissertation aims to simplify the above-mentioned pronunciation and conjugation of Portuguese verbs by creating a website focused on this subject. This website allows to conjugate each verb at both orthographic and phonetic ways. Moreover, it is also possible to hear these verbs by using a speech synthesis system.

To create the speech synthesizer, an HMM was used (HTS). The HTS creates the HMM models to further be used in speech synthesis. To create the HMM models, the HTS needs to perform a training based on a database of audio files. These audio files contain records of speech from two different speakers - a male and a female ones.

After training process, files describing the synthetic speech are created. These files are then used on the website, allowing the users to listen the pronunciation of the verbs. The website users are able to conjugate each verb and listen its respective pronunciation. The conjugation and pronunciation mechanisms of inflections are made by using two C++ applications. If an introduced verb on the website makes part of the previously audio files recorded by the speakers, there is no speech synthesis. In these cases, the original file is directly used to play the verb. Finally, the website also allows that any other verbs can be conjugated even if they exist or if they don't exist.

Keywords

HMM, HTS, Synthesis, Verbs, Website

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

AC	Ana Coelho
API	Application Program Interface
ASCII	American Standard Code for Information Interchange
CSS	Cascading Style Sheets
DFT	Discrete Fourier Transform
HMM	Hidden Markov Models
HTML	HyperText Markup Language
HTS	HMM-based Speech Synthesis System
IIS	Internet Information Services
IPA	International Phonetic Alphabet
JC	João Constantino
JS	JavaScript
MFCC	Mel-Frequency Cepstral Coefficients
MGC	Mel Generalized Cepstral
MLF	Master Label File
MLSA	Mel Log Spectral Approximation
PDF	Probability density function
SAMPA	Speech Assessment Methods Phonetic Alphabet

TTS Text-To-Speech

URLs Uniform Resource Locators

USB Universal Serial Bus

Capítulo 1

Introdução

1.1 Motivação e objetivo

Segundo o estudo do Instituto Superior de Ciências do Trabalho e da Empresa (ISCTE/IUL), a língua portuguesa é a quarta mais falada do mundo registando uma das taxas de crescimento mais elevadas na Internet, nas redes sociais e na aprendizagem como língua estrangeira [1]. A língua portuguesa é composta por inúmeras regras e elementos que permitem aos seus falantes comunicar entre si. Um dos elementos mais importantes na aprendizagem da língua portuguesa (e outras em geral), é o domínio dos verbos e respetivas flexões. Os verbos em português são flexionados em pessoa, número, tempo e modo, apresentado ainda entre si, algumas irregularidades. Este facto torna o processo de aprender a conjugar e pronunciar verbos bastante árduo para os falantes, principalmente para aqueles cujo português não é a sua primeira língua.

Recorrendo à tecnologia atual, podem ser desenvolvidas ferramentas vocacionadas para auxiliar na aprendizagem da língua portuguesa, tornando o processo mais dinâmico para os falantes. É neste contexto que surge o tema desta dissertação, a elaboração de um sistema *online* capaz de conjugar e pronunciar verbos em português. Embora existam já diversas soluções na internet que fornecem aos utilizadores flexões verbais em forma ortográfica, estas não apresentam informação acerca da sua fonética, ou, quando apresentada, esta mostra ser bastante incompleta. O objetivo desta dissertação consiste em colmatar esse problema, fornecendo uma ferramenta prática onde os utilizadores dispõem de flexões ortográficas e fonéticas, tendo também a opção de ouvir, através de síntese de fala, como se pronuncia qualquer flexão verbal. Para tal é necessário recorrer a sistemas de síntese de fala, conhecidos vulgarmente como sistemas Text-To-Speech (TTS). Estes sistemas visam a produção de uma fala sintética a

partir de um texto fornecido [2]. As qualidades mais importantes de um sistema TTS são a naturalidade (semelhança com a fala humana) e inteligibilidade da fala resultante [2]. Estes sistemas podem ter como base diferentes técnicas. Destacam-se, por serem das mais utilizadas, a síntese baseada em concatenação de segmentos de fala e a síntese baseada em HMMs [3].

Nesta dissertação utilizou-se o sistema HTS [5], um sistema de síntese baseado em HMMs que desde o seu lançamento em 2002, tem vindo a conquistar o seu lugar na área da síntese de fala, sendo hoje utilizado por empresas como a Microsoft e IBM [6].

Esta dissertação tem como referência vários trabalhos anteriores [7][8] onde o tema de síntese de fala recorrendo ao HTS está presente. Estes têm vindo a utilizar e melhorar o HTS para o treino de modelos de fonemas da língua portuguesa. Ao contrário dos projetos anteriores, o treino de fala nesta dissertação é realizado apenas com locuções de conjugações verbais, sendo suficientemente ricas foneticamente para permitir a pronunciação de qualquer verbo.

1.2 Fonologia e fonética da língua portuguesa

A linguagem humana é um dos sistemas de comunicação existentes mais complexos. Para o utilizar, é necessário possuir um bom conhecimento de uma determina língua, um processo moroso adquirido ao longo de vários anos. A área responsável pelo estudo científico da linguagem designa-se por linguística, onde se encontram inseridas a fonologia e a fonética. Apesar de serem áreas relacionadas, estas têm um foco de estudo diferente. A fonologia é o ramo da linguística responsável pelo estudo da organização sistemática de sons numa linguagem, classificando-os em unidades designadas por fonemas (unidades sonoras mínimas numa dada língua)[1]. Analisando a pronunciação das palavras $\langle vaca \rangle$ e $\langle faca \rangle$, verifica-se que estas diferem apenas nos fonemas [v] e [f], alterando assim o seu significado. A fonética por outro lado, estuda a natureza física da produção e perceção dos sons da fala humana. As unidades básicas utilizadas na fonética são denominadas por fones[1].

De forma a representar os fonemas da fala humana, utilizam-se alfabetos produzidos para o efeito, os alfabetos fonéticos. Um dos alfabetos fonéticos mais utilizados é o IPA [9] representado na tabela 1.1.

Embora o alfabeto IPA seja o mais utilizado, em aplicações informáticas não se torna prática a sua implementação, uma vez que este não é composto apenas por caracteres ASCII, dificultando, assim, a introdução dos seus símbolos recorrendo a um teclado normal. Para colmatar este problema, é

Símbolo IPA	Palavra	Transcrição fonética	Símbolo IPA	Palavra	Transcrição fonética
ɐ	assim	ɐsĩ	d	dado	dadu
a	arma	armɐ	g	gato	gatu
ə	está	ɘʃta	p	pato	patu
e	eu	ew	t	sapato	sɐpatu
ɛ	hélio	ɛliu	k	quatro	kuatru
i	livro	livru	f	filmar	filmar
o	outro	otru	s	sal	sal
ɔ	óculos	ɔkuluʃ	ʃ	chaves	ʃavɘʃ
u	luva	luvɐ	v	vós	vɔʃ
ɐ̃	canto	kɐ̃tu	z	zebra	zebrɐ
ẽ	ênfase	ẽfezɘ	ʒ	joia	ʒɔjɐ
ĩ	cinto	sĩtu	l	letra	letrɐ
õ	ontem	õtɐ̃ʃ	ʎ	milha	miʎɐ
ũ	umbigo	ũbigu	ɾ	arar	ɐrar
j	caixa	kajʃɐ	ʁ	rato	ʁatu
w	pauta	pawtɐ	m	matar	mɐtar
j̃	cães	kɐ̃j̃ʃ	n	nadar	nɐdar
w̃	não	nɐ̃w̃	ɲ	desenho	dɘzɐɲu
b	barco	barku			

Tabela 1.1: Símbolos e exemplos do alfabeto IPA. *Adaptado de [10]*

utilizado frequentemente um outro alfabeto fonético, o Speech Assessment Methods Phonetic Alphabet (SAMPA) [11], ilustrado na tabela 1.2.

1.3 Verbos da língua portuguesa

Na gramática da língua portuguesa, uma palavra pode enquadrar-se morfologicamente dentro de várias classes [12], entre elas verbos.

Esta dissertação foca-se na classe de verbos, tornando-se então imperativo a realização de uma pequena introdução acerca deste tema. Os verbos são palavras que indicam ações, qualidades ou estados, podendo variar em modo, tempo e número. Na gramática portuguesa, os modos e tempos verbais são os seguintes:[10]

- Indicativo: presente; pretérito imperfeito; pretérito perfeito; pretérito mais-que-perfeito; futuro do presente; futuro do pretérito (condicional)
- Conjuntivo: presente; pretérito imperfeito; futuro
- Imperativo

Simbolo SAMPA	Palavra	Transcrição fonética	Simbolo SAMPA	Palavra	Transcrição fonética
i	vinte	vi~t@	**t**	tenho	t6Ju
e	fazer	f6zer	**d**	doce	dos@
e	belo	bElu	**k**	com	ko~
a	falo	falu	**g**	grande	gr6~d@
6	cama	k6m6	**f**	falo	falu
O	ontem	O~t6~j~	**v**	verde	verd@
o	lobo	lobu	**s**	céu	sEw
u	jus	ZuS	**z**	casa	kaz6
@	felizes	f@liz@S	**S**	chapéu	S6pEw
i~	fim	fi~	**Z**	joia	ZOj6
e~	emprego	e~pregu	**m**	mar	mar
6~	irmã	irm6~	**n**	nada	nad6
o~	bom	bo~	**J**	vinho	viJu
u~	um	u~	**l**	lanche	l6~S@
j~	mãe	m6~j~	**L**	trabalho	tr6baLu
w~	cão	k6~w~	**r**	caro	karu
p	pai	paj	**R**	rua	Ru6
b	barco	barku			

Tabela 1.2: Símbolos e exemplos do alfabeto SAMPA. *Adaptado de [11]*

Existem ainda três formas nominais dos verbos:

- Infinitivo
- Infinitivo Pessoal
- Gerúndio
- Particípio passado

Em termos de variação de número, existem três pessoas do singular (eu, tu, ele) e três pessoas do plural (nós, vós, eles).

À variação das formas verbais é dado o nome de flexões que, agrupadas de forma ordenada em todos os modos, tempos, pessoas e números, constituem a conjugação de um verbo[13].

Quanto à sua estrutura, os verbos possuem radical (a parte invariável), terminação (parte flexionada) e a vogal temática, que caracteriza a conjugação. Existem três conjugações na gramática portuguesa:[12]

- 1ª Conjugação: verbos com vogal temática -a- (terminados em ar)
- 2ª Conjugação: verbos com vogal temática -e- (terminados em er)
- 3ª Conjugação: verbos com vogal temática -i- (terminados em ir)

Há ainda que destacar, a existência do verbo pôr e seus derivados que, devido às suas irregularidades não se enquadram em qualquer classificação.

Quando um verbo se flexiona de acordo com o paradigma da conjugação, designa-se por verbo regular, caso contrário por irregular[13].

A maior parte dos verbos são regulares tanto na sua grafia como na sua pronunciação, sempre que as flexões têm terminações regulares. Nos verbos regulares, a silaba tónica está presente, na maioria dos casos, nas terminações (formas arrizotónicas) enquanto que nos verbos irregulares reside no radical (formas rizotónicas). Às variações vocálicas na pronunciação das flexões verbais é dado o nome de irregularidades de pronunciação. Os diferentes tipos de irregularidades de pronunciação foram estudados de forma detalhada, com o objetivo de estabelecer um conjunto de paradigmas de pronunciação [10]. Estes refletem uma escolha de um verbo tipo, existindo muitos outros que seguem o mesmo tipo de irregularidade na pronunciação. Na tabela 1.3 é possível identificar os 57 paradigmas de pronunciação para o português europeu, que se encontram ordenados segundo uma escala de irregularidade estabelecida em [10], tendo como base 3 fatores: o número de flexões irregulares na pronunciação (nIP tabela 1.3) o número de alterações no radical e o número de alterações nos sufixos.

Nesta dissertação, uma parte importante do trabalho desenvolvido consiste na deteção automática do paradigma do verbo a pronunciar, onde se utilizou um conjunto de regras para o efeito. Este assunto é abordado na secção 4.2.

	Verbo	Pronunciação	nIP		Verbo	Pronunciação	nIP
Regulares	amar	ɐmˈar	0		ferir	fɔrˈir	11
	viver	vivˈer	0		requerer	ʁɔkɔrˈer	11
	unir	unˈir	0		jazer	ʒɐzˈer	11
	pôr	pˈor	0		roer	ʁuˈer	9
Quase-regulares	induzir	ĩduzˈir	2		escrever	ɐʃkrɔvˈer	10
	dormir	durmˈir	4		medir	mɔdˈir	11
	erguer	ergˈer	4		perder	pɔrdˈer	11
	aquece	ɐkˈɛsɔ	5		valer	vɐlˈer	11
	afluir	ɐfluˈir	4		trair	trɐˈir	11
	construir	kõʃtruˈir	4		rir	ʁˈir	13
	cobrir	kubrˈir	5		ler	lˈer	13
	ouvir	ovˈir	7		prover	pruvˈer	13
	refletir	ʁɔflɛtˈir	7	Irregulares	aprazer	ɐprɐzˈer	33
	sentir	sẽtˈir	7		caber	kɐbˈer	35
	debate	dɔbˈatɔ	9		querer	kɔrˈer	35
	desejar	dɔzɔʒˈar	9		saber	sɐbˈer	35
	desenhar	dɔzɔɲˈar	9		poder	pudˈer	35
	errar	iʁˈar	9		dar	dˈar	31
	lavar	lɐvˈar	9		haver	ɐvˈer	35
	somar	sumˈar	9		ver	vˈer	37
	agir	ɐʒˈir	9		estar	ɐʃtˈar	34
	chegar	ʃɐgˈar	9		ter	tˈer	43
	negar	nɐgˈar	9		ir	ˈir	38
	tocar	tukˈar	9		vir	vˈir	44
	ansiar	ẽsjˈar	9		dizer	dizˈer	46
	beber	bɐbˈer	9		trazer	trɐzˈer	47
	mover	muvˈer	9		ser	sˈer	44
	agredir	ɐgrɐdˈir	11		fazer	fɐzˈer	48
	abrir	ɐbrˈir	10				

Tabela 1.3: Lista dos 57 paradigmas de irregularidades na pronunciação dos verbos. *Adaptado de [10]*

1.4 Modelos de Markov não observáveis (HMM)

Os modelos de Markov não observáveis são modelos largamente utilizados para modelar os parâmetros da fala humana, tendo sido aplicados, com enorme sucesso, em sistemas de reconhecimento de fala. O sistema HTS utilizado nesta dissertação utiliza também HMMs como unidades de fala.

Um HMM é uma máquina de estados finita que gera uma sequência de observações temporais discretas, representadas por um estado. Os HMMs baseiam-se na cadeia de Markov, onde para cada

estado da cadeia, é definida uma função densidade de probabilidade - Probability density function (PDF), que descreve o sinal observado nesse estado. Devido ao facto da sequência de estados implícita numa sequência de observações não ser conhecida, estes modelos são então designados por modelos de Markov não observáveis. O sinal de fala, devido às constantes variações de propriedades no tempo, não é um sinal estacionário. Contudo, ao realizar-se uma análise de tempo curto, isto é, considerando pequenos intervalos de tempo para análise do sinal de fala, este pode ser considerado estacionário, além disso, a fala pode ser vista como uma sequência discreta de fones, podendo então ser modelada por um HMM. Em cada intervalo de tempo, existe uma transição de estados na cadeia de Markov, de acordo com a matriz de probabilidades de transição de estado. Em cada transição, são gerados dados de observação (**o**), em consonância com a função densidade de probabilidade do estado atual. O HMM é então um modelo duplamente estocástico. Um HMM de ordem N é definido pela matriz de probabilidades de transição $\mathbf{A} = \left\{a_{ij}^{N}\right\}$, sendo a_{ij} a probabilidades de transição do estado i para o estado j; pelas N funções densidade de probabilidade dos estados $\mathbf{B} = \left\{b_i(\mathbf{o})^N\right\}$. No caso de modelos ergódicos (onde cada estado pode transitar para qualquer outro estado, ver figura 1.1a)) é necessário ainda definir as probabilidades de ocupação inicial dos estados $\prod = \left\{{\pi_i}^N\right\}$. A função densidade de probabilidade de um estado i com observações **o** é normalmente modelada por uma mistura de distribuições Gaussianas, dada pela expressão:

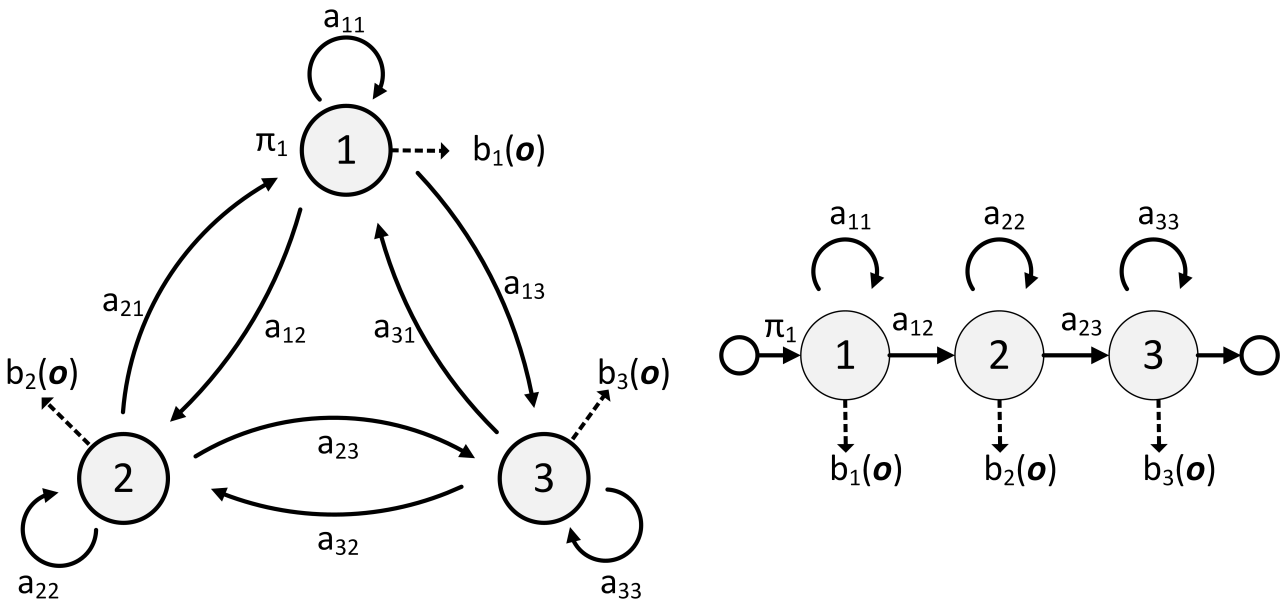
$$b_i(\mathbf{o}) = \sum_{m=1}^{M} w_{im}\mathcal{N}(\mathbf{o}; \mu_{im}, \Sigma im), \tag{1.1}$$

com M o número de componentes da mistura para a distribuição, w_{im} o peso dessa componente na mistura, μ_{im} o vetor de médias (de dimensão $L \times 1$) e Σim uma matriz de covariâncias $L \times L$ do componente da mistura m do estado i. A distribuição Gaussiana $\mathcal{N}(\mathbf{o}; \mu_{im}, \Sigma im)$ de cada componente é definida por:

$$\mathcal{N}(\mathbf{o}; \mu_{im}, \Sigma im) = \frac{1}{\sqrt{(2\pi)^L \left|\Sigma_{im}\right|}} exp\left(-\frac{1}{2}(\mathbf{o} - \mu_{im})^T \Sigma_{im}^{-1}(\mathbf{o} - \mu_{im})\right), \tag{1.2}$$

com L a dimensão do vetor de observações.

A figura 1.1 mostra dois exemplos da estrutura típica de um HMM. A figura 1.1a representa um modelo ergódico com três estados, no qual cada estado pode alcançar todos os outros estados com uma única transição. A figura 1.1b representa um modelo esquerda-direita com cinco estados no total. Três desses estados são emissores e têm uma função densidade de probabilidade associada. O primeiro e o último estado são designados por estados não emissores, servindo apenas para concatenação de HMMs.



(a) modelo ergódico

(b) modelo esquerda-direita

Figura 1.1: Exemplo da estrutura de um HMM. *Adaptado de [14]*

Neste modelo, um estado pode transitar apenas para o estado à sua direita, ou manter-se no estado atual. Os modelos esquerda-direita são normalmente utilizados como unidades de modelação dos parâmetros da fala. No HTS, o modelo utilizado é o modelo esquerda-direita. Este é composto por sete estados (cinco emissores e dois não emissores) no caso da frequência fundamental (tom) e do espetro, mas composto por três estados (um emissor e dois não emissores) no caso da duração de cada fone [14].

1.5 Sistema de síntese de fala baseado em HMMs (HTS)

O HTS foi desenvolvido no Departamento de Ciências Computacionais do Instituto de Tecnologia de Nagoya no Japão, tendo sido lançada a sua primeira versão em 2002 e vindo a ser atualizado até à data. Este sistema apresenta-se como uma alteração ao sistema Hidden Markov Model Toolkit (HTK) [15], ferramenta bastante utilizada no reconhecimento de fala. O HTS permite então utilizar para síntese de fala, as mesmas ferramentas que o HTK [5]. O funcionamento geral do HTS pode ser dividido em duas etapas,a fase de treino e a fase de síntese, como ilustrado na figura 1.2.

Na fase de treino é utilizada uma base de dados de fala, composta pelas gravações de um locutor. Esta deve ter sempre uma grande variedade fonética, boa qualidade de áudio e uma prosódia constante,

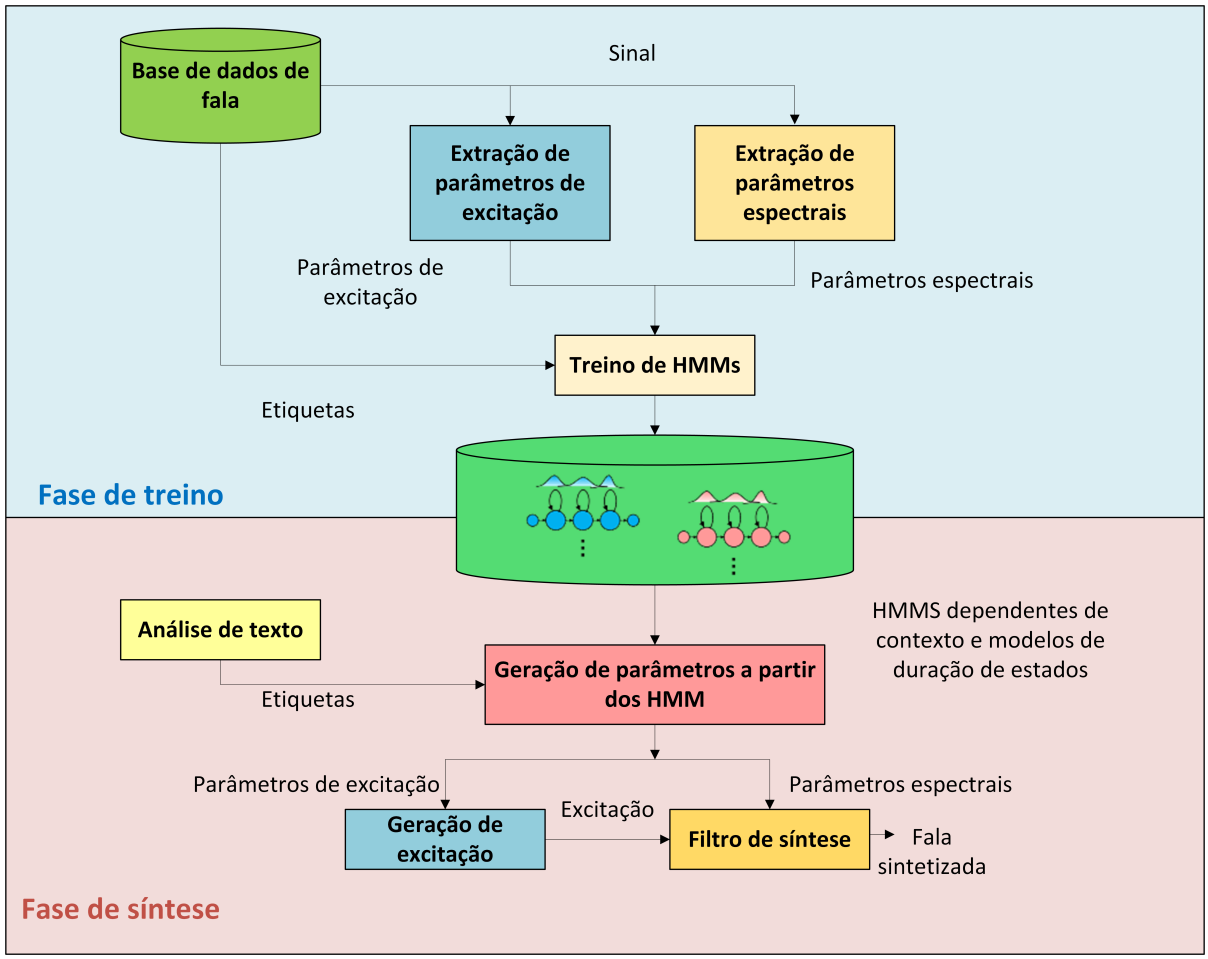


Figura 1.2: Esquema geral do sistema HTS. *Adaptado de [6]*

uma vez que, é uma parte fundamental para atingir bons resultados no final do treino. A partir da base de dados são retirados os parâmetros espectrais e de excitação que, juntamente com os ficheiros de etiquetas de fones das locuções gravadas, são utilizados na fase de treino para a criação dos modelos HMM, que serão utilizados na fase da síntese. Existem dois tipos de etiquetas de fones; as etiquetas que não apresentam contexto fonético, referindo-se apenas ao fone em causa (conhecidas como etiquetas de **monofones**) e as que possuem contexto fonético, que se referem não só ao fone em causa, mas também a fones vizinhos a este. O sistema HTS utiliza etiquetas de monofones e etiquetas de **pentafones**, isto é, etiquetas que consideram para além do fone atual (fone central) os dois fones à sua esquerda e os dois fones à sua direita. As etiquetas de pentafones têm também presentes informação contextual, sintática e gramatical. Por vezes são também definidos **trifones**, fones com indicação dos fones à esquerda e à direita do fone em questão.

Na fase de síntese, o sistema recebe o texto a sintetizar e cria as etiquetas de pentafones correspondentes. Baseado nestas etiquetas, é construida uma sequência HMM, concatenando vários modelos HMM resultantes da fase de treino. Desta sequência HMM são obtidos os parâmetros espectrais e de

excitação e, por fim, um ficheiro de áudio é sintetizado usando um filtro Mel Log Spectral Approximation (MLSA) [16]. A utilização deste filtro advém da sensibilidade à frequência do ouvido humano não ser uma função linear, ou seja, o ouvido humano tem maior sensibilidade para baixas frequências do que para frequências altas, sendo necessário transformar a frequência linear f numa escala não linear, como a escala de Mel (figura 1.3) [14].

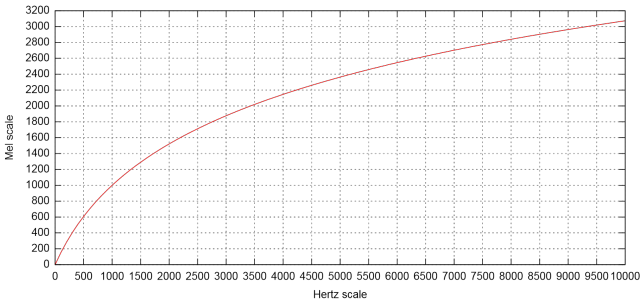


Figura 1.3: Escala de Mel vs Hertz. *Retirado de [17]*

A figura 1.4 representa um modelo para a produção de fala humana. No HTS, a função transferência do filtro $H(z)$ é dada por:

$$H(z) = exp\left[\sum_{m=0}^{M} c(m)\tilde{z}^{-m}\right] \tag{1.3}$$

onde $c(m)$ é o vetor de coeficientes espectrais de tamanho M+1 e $\tilde{z} = \left[1, \tilde{z}^{-1}, ..., \tilde{z}^{-M}\right]^T$.O sistema $\tilde{z}^{-1}$ é definido pela seguinte função passa tudo de primeira ordem:

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, |\alpha| < 1 \tag{1.4}$$

que faz a aproximação da escala em frequência para a escala de Mel e, onde α corresponde ao fator de distorção.

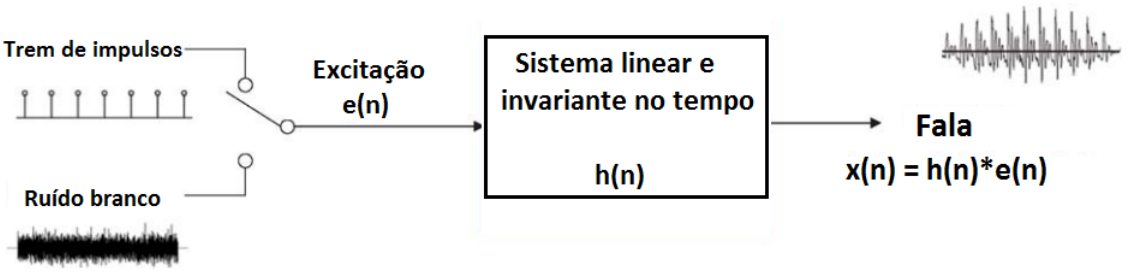


Figura 1.4: Modelo do filtro que simula a fala humana. *Adaptado de [4]*

A frequência distorcida é dada por :

$$\beta(\omega) = \tan^{-1} \frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha} \tag{1.5}$$

Na figura 1.5 está representada a distorção da frequência provocada pelo filtro passa tudo, onde se pode verificar que, para a frequência de amostragem utilizada pelo HTS (48kHz) é de 0.55 [14].

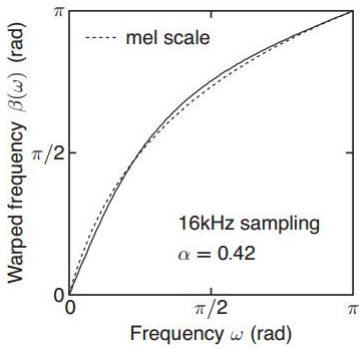


Figura 1.5: Distorção de frequência pelo filtro passa tudo. *Retirado de [14]*

Para se realizar o treino de uma forma correta, é necessário utilizar as ferramentas do HTS seguindo uma ordem lógica de implementação. Este processo é facilitado graças ao fornecimento de *demos* juntamente ao código fonte do HTS. Nesta dissertação foi utilizada uma *demo* de um trabalho anterior. Esta *demo* sofreu as alterações necessárias para o treino de modelos do português europeu, uma vez que, originalmente, estava destinada ao treino de modelos para o português brasileiro. A *demo* é composta por *scripts* em Perl e Tcl que recorrem a ferramentas como o Speech Signal Processing Toolkit (SPTK) [18] e Sound eXchange (SoX) [19] para a análise espetral , extração de tom e manipulação dos ficheiros áudio. O processo de treino irá ser abordado com maior detalhe no capitulo 3.

Capítulo 2

Base de dados

Tal como supra ilustrado na figura 1.2, a primeira etapa num sistema de síntese envolve uma base de dados de fala. É então necessário construir a base de dados recolhendo locuções de um ou vários locutores. A recolha das locuções deve ser feita preferencialmente, num ambiente ausente de ruídos externos e com boas características acústicas. Os locutores devem manter um tom e uma entoação constante ao longo das gravações e ter uma boa dicção, sem cometer hesitações ou erros na pronunciação. Uma boa base de dados de fala é um fator extremamente importante para o sucesso do sintetizador. Os subcapítulos seguintes descrevem o processo da construção da base de dados utilizada nesta dissertação.

2.1 Frases para gravação

A escolha do conjunto de frases para as gravações deve ter em conta o seu conteúdo fonético que, deve ser o mais diversificado possível, levando ao aumento do número de modelos treinados, e, por conseguinte a qualidade do sistema. Uma vez que, o objetivo do projeto é a construção de um sintetizador de fala para conjugar verbos, as frases escolhidas para as gravações são constituídas pela conjugação dos 57 verbos paradigma do português europeu, em todos os tempos, modos e formas nominais dos verbos (figura 2.1), resultando num total de 627 frases. Para que o conteúdo fonético da base de dados pudesse ser um pouco mais alargado, foram adicionadas mais algumas frases, contendo apenas o infinitivo e as duas primeiras pessoas do presente do indicativo (figura 2.2), totalizando 1583 frases.

```
amar eu amo tu amas ele ama nós amamos vós amais eles amam
eu amava tu amavas ele amava nós amávamos vós amáveis eles amavam
eu amei tu amaste ele amou nós amámos vós amastes eles amaram
eu amara tu amaras ele amara nós amáramos vós amáreis eles amaram
eu amarei tu amarás ele amará nós amaremos vós amareis eles amarão
```

Figura 2.1: Excerto do *prompt* de gravação 1

```
consolidar eu consolido tu consolidas
confiar eu confio tu confias
acelerar eu acelero tu aceleras
implicar eu implico tu implicas
testar eu testo tu testas
confundir eu confundo tu confundes
```

Figura 2.2: Excerto do *prompt* de gravação 2

2.2 Gravação das locuções

2.2.1 AudioPrompt

Tendo definidas as frases, é necessário um *software* de apoio que facilite o processo de gravação. *AudioPrompt* é uma aplicação que foi desenvolvida no laboratório no âmbito de um projeto anterior, tendo vindo a ser atualizada ficando cada vez mais completa.

Figura 2.3: Menu inicial do *AudioPrompt*

Ao iniciar-se uma sessão de gravação, o utilizador deve introduzir a informação do locutor, a lista de frases a gravar, assim como, as definições que pretende utilizar na sessão. Para a gravação das locuções nesta base de dados foi utilizada uma frequência de amostragem de 48kHz.

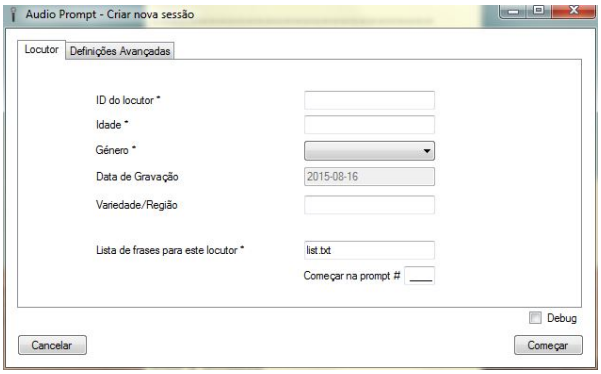


Figura 2.4: Ecrã de nova sessão do *AudioPrompt*

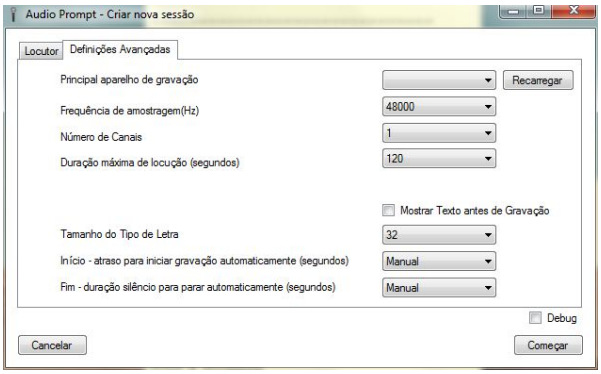


Figura 2.5: Ecrã de definições avançadas *AudioPrompt*

Após introduzir os dados pretendidos, o programa está pronto para iniciar a gravação. É apresentado um ecrã onde o utilizador dispõem de todos os controlos necessários para realizar as gravações.

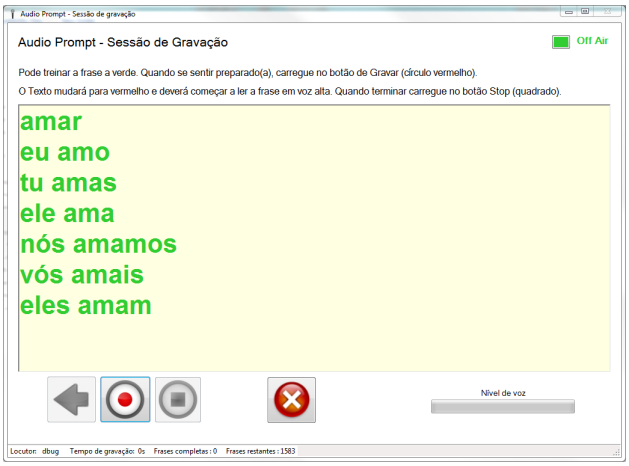


Figura 2.6: Ambiente de gravação do *Audioprompt*

2.2.2 Material utilizado

O material utilizado na gravação das locuções foi o seguinte:

- Câmara de gravação não reverberante (figura 2.7)
- Computador portátil
- Microfone de estúdio Rode NT1-A
- Microfone USB Samson GoMic
- Placa de som Creative EMU 0404 USB

O computador portátil, que corria a aplicação *AudioPrompt*, tinha ligado a si a placa de som Creative EMU e o microfone Universal Serial Bus (USB) Samson GoMic. A utilização da placa de som possibilita a conversão de sinal analógico, recebido do microfone de estúdio, para digital e, por consequência, a utilização do microfone na aplicação. O facto do sistema de gravação ser composto por dois microfones cobre o risco de se ter uma locução vazia devido a alguma falha técnica que possa ocorrer. Uma demonstração do sistema de gravação completo encontra-se na figura 2.8.

Figura 2.7: Câmara de gravação

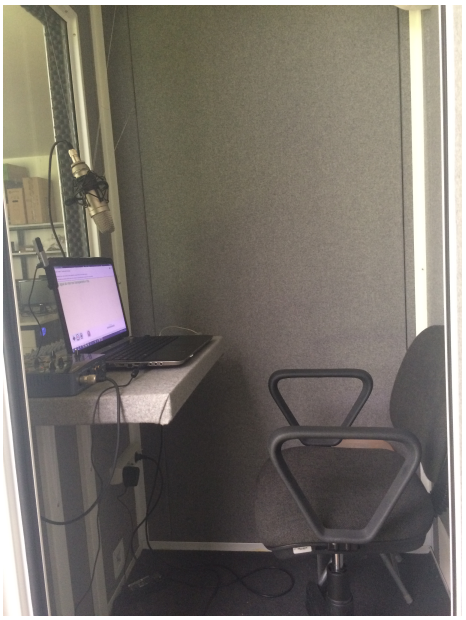


Figura 2.8: Sistema de gravação

2.2.3 Gravações e locutores

Inicialmente foi estabelecido que um dos critérios do *website*, seria a possibilidade dos utilizadores ouvirem a pronunciação dos verbos com uma voz masculina e outra feminina, sendo portanto, necessária a colaboração de dois locutores para o desenvolvimento do projeto (figura 2.9 e 2.10). Para que não ocorressem variações nas características das vozes dos locutores, cada locutor realizou as sessões num único dia. Devido às características da câmara de gravação, sessões com uma duração muito elevada levam a um aumento da temperatura no interior desta, tornando-se desconfortável para o locutor. Foram então realizadas sessões de aproximadamente uma hora, fazendo-se intervalos para descanso do locutor e arrefecimento da sala. Durante os intervalos ambos os microfones eram testados, assim como, as últimas locuções gravadas, prevenindo assim qualquer erro que pudesse surgir. Para concluir as gravações das 1583 frases foram necessárias aproximadamente 6 horas.

Figura 2.9: Gravações Ana Coelho (AC)

Figura 2.10: Gravações João Constantino (JC)

2.3 Tratamento das locuções

Terminadas as sessões de gravação, o resultado final foram duas bases de dados com 1583 locuções cada, resultando em aproximadamente duas horas e meia de fala para cada locutor. Para se eliminarem alguns ruídos de baixa frequência (vibrações da mesa de apoio ou sopros), foi aplicado, recorrendo a um *script Matlab* (figura 2.11) um filtro Chebychev de tipo II passa alto com frequência de corte de 150 Hz.

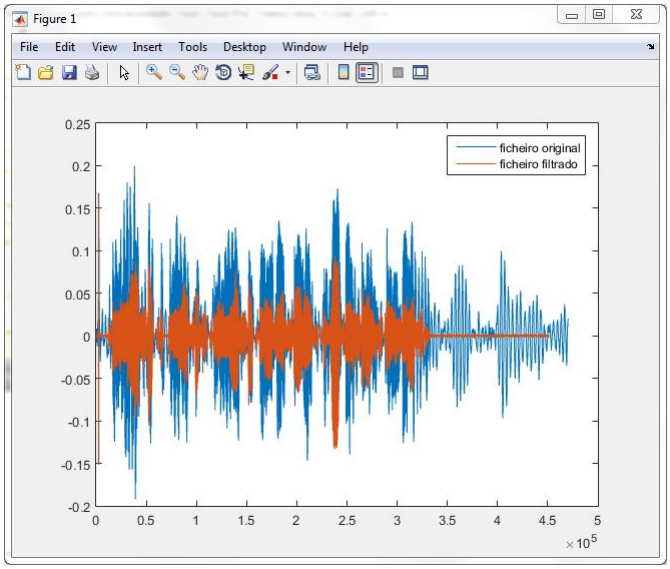


Figura 2.11: Exemplo de filtragem de ficheiros wav

Após a filtragem, a energia do sinal é calculada em tramas de 100 ms, o que permite analisar o sinal

e identificar momentos de silêncio. Definiu-se como limite máximo um intervalo de 0.5s de silêncio, pelo que, caso o sinal possua no seu inicio ou final intervalos superiores a este, o excesso de silêncio é eliminado.

Como referido anteriormente, as locuções gravadas correspondem a tempos, modos ou formas nominais completas. Sendo o sintetizador utilizado para pronunciar verbos, este deve conseguir sintetizar com sucesso um tempo verbal completo, assim como a flexão de uma única pessoa. Embora o sistema de treino com as 1583 locuções gravadas conseguisse os modelos suficientes para essa síntese, decidiu fazer-se o corte individual das locuções correspondentes aos verbos paradigma, para todas as pessoas. Este procedimento, embora trabalhoso, revelou ter uma grande contribuição para a qualidade final do sintetizador. Para se fazer o corte das locuções foi utilizado o *software* Audacity [20], um editor de áudio *open-source* (figura 2.12). No final do corte das locuções, as bases de dados passaram a ser compostas por 5288 ficheiros.

Após o tratamento das locuções, segue-se o processo de alinhamento das transcrições fonéticas com o sinal de fala, que será referido na secção 3.1.

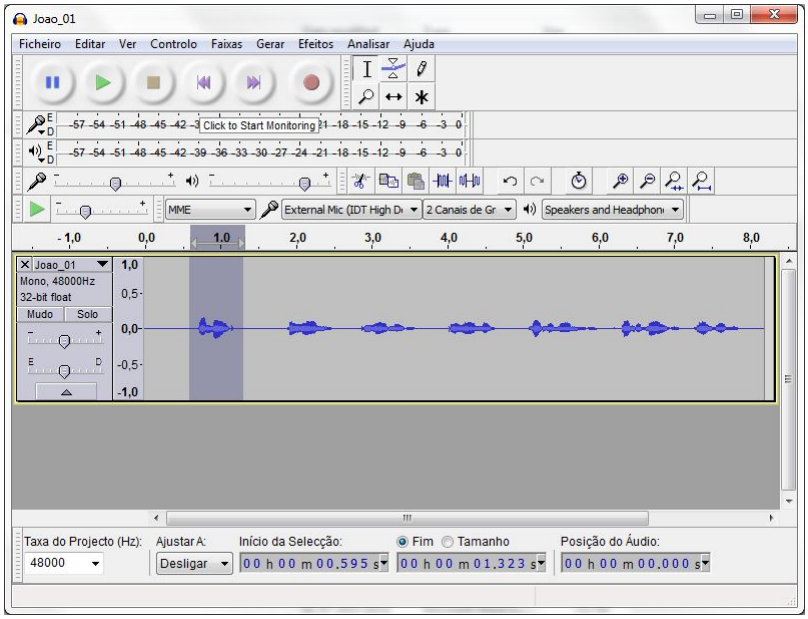


Figura 2.12: Corte de ficheiros wav com o *software Audacity*

Capítulo 3

Treino de voz

Para a realização do treino de uma voz [1], a *demo* HTS necessita dos seguintes ficheiros de entrada:

- Ficheiros de áudio em formato *raw* [2] a 48kHz.
- Ficheiro de etiquetas de monofones alinhados com os ficheiros áudio.
- Ficheiros de etiquetas de pentafones alinhados com os ficheiros áudio.
- Ficheiro de questões para a criação de árvores de decisão.

Os ficheiros de áudio correspondem às 5288 locuções que constituem a base de dados de cada locutor. Estes ficheiros foram convertidos para formato *raw* a 48kHz e para formato *wav* a 16kHz utilizando a ferramenta SoX [19] para poderem ser utilizados no treino e no alinhamento, respetivamente.

O ficheiro de questões utilizado nesta dissertação foi criado no âmbito de um projeto anterior, onde se adaptou para o caso do português europeu [7].

Os ficheiros de etiquetas de monofones e pentafones resultam de uma etapa anterior ao treino denominada de alinhamento.

[1] Entende-se por voz o conjunto de parâmetros do sintetizador treinados para um dado locutor.
[2] Ficheiros de áudio não comprimidos e sem cabeçalho.

3.1 Alinhamento

O alinhamento é o processo onde se alinham temporalmente os ficheiros áudio da base de dados com a transcrição fonética das frases gravadas. Para realizar esse alinhamento foram utilizadas algumas ferramentas do HTK [15]. Como a etapa de alinhamento iria ser realizada para dois locutores, foi feito um *script* em *Matlab* que permite realizar esta tarefa múltiplas vezes de forma simples.

A primeira tarefa a ser realizada é a transcrição fonética das frases de gravação 3.1. Aqui, utilizou-se o mesmo processo que na criação das próprias frases, recorrendo-se a um *script* em *Matlab* desenvolvido no laboratório, que, dado o infinitivo de um verbo e a sua transcrição fonética conjuga-o ortográfica e foneticamente nos tempos pretendidos. A transcrição fonética dos verbos paradigma foi retirada da tabela 1.3, enquanto que, nos restantes verbos foi utilizado um conversor de grafemas para fonemas existente no laboratório denominado g2p [21]. O alfabeto fonológico utilizado nesta etapa é o SAMPA, devido a este utilizar caracteres ASCII para representar os fonemas, permitindo assim serem introduzidos por um teclado de computador normal. Os ficheiros contendo a lista de frases e a sua transcrição fonética são assim alvo de um processamento, do qual resulta um dicionário contendo todas as palavras (não repetidas) e a sua correspondente transcrição fonética (figura 3.2).

```
1  6m"ar "ew "6mu t"u "6m6S "el@ "6m6 n"OS 6m"6muS v"OS 6m"ajS "el@S "6m6~w~
2  "ew 6m"av6 t"u 6m"av6S "el@ 6m"av6 n"OS 6m"av6muS v"OS 6m"av6jS "el@S 6m"av6~w~
3  "ew 6m"6j t"u 6m"aSt@ "el@ 6m"o n"OS 6m"amuS v"OS 6m"aSt@S "el@S 6m"ar6~w~
4  "ew 6m"ar6 t"u 6m"ar6S "el@ 6m"ar6 n"OS 6m"ar6muS v"OS 6m"ar6jS "el@S 6m"ar6~w~
5  "ew 6m6r"6j t"u 6m6r"aS "el@ 6m6r"a n"OS 6m6r"emuS v"OS 6m6r"6jS "el@S 6m6r"6~w~
6  "ew 6m6r"i6 t"u 6m6r"i6S "el@ 6m6r"i6 n"OS 6m6r"i6muS v"OS 6m6r"i6jS "el@S 6m6r"i6~w~
7  k@ "ew "6m@ k@ t"u "6m@S k@ "el@ "6m@ k@ n"OS 6m"emuS k@ v"OS 6m"6jS k@ "el@S "6m6~j~
```

Figura 3.1: Excerto da transcrição fonética das frases. As aspas indicam que a vogal seguinte está em posição tónica.

```
1  abalar   ª b ª l á r
2  abalas   ª b á l ª S
3  abalo    ª b á l u
4  abandonar   ª b ã d u n á r
5  abandonas   ª b ã d ô n ª S
6  abandono    ª b ã d ô n u
7  abastecer   ª b ª S t @ s ê r
```

Figura 3.2: Excerto do dicionário contendo todas as palavras e a sua correspondente transcrição fonética. Nesta transcrição o símbolo "6"é substituído por "ª"e a indicação de tónica e de nasal é aglutinada com a vogal com um diacrítico (por exemplo <á> em vez de <"a> e <ã> em vez de <a∼>).

Para realizar o alinhamento é necessário retirar os coeficientes espectrais dos ficheiros áudio. Para

tal, utiliza-se a ferramenta *HCopy* que cria ficheiros Mel-Frequency Cepstral Coefficients (MFCC) [3] a partir dos ficheiros áudio em formato *wav* com uma frequência de amostragem de 16kHz.

De seguida utiliza-se a ferramenta *HDMan* para criar uma lista de possíveis sequencias de três fones (trifones) e um dicionário de trifones a partir do dicionário criado anteriormente.

É ainda necessário utilizar a ferramenta *HHed* que recorre à lista de trifones e a um ficheiro existente no laboratório, que contém alguns modelos HMM gerados previamente, para criar os modelos HMM de trifones presentes nas frases de gravação.

Por fim, a ferramenta *HVite* utiliza os modelos criados para realizar o alinhamento. O resultado, é um ficheiro Master Label File (MLF) [22] que contém o alinhamento temporal de todas as frases com o áudio correspondente (figura 3.3). Os ficheiros de etiquetas de monofones são obtidos separando o ficheiro MLF em ficheiros individuais [22].

```
1  #!MLF!#
2  "*/Ana_01_2.lab"
3  0   4000000 sil sil
4  4000000 5300000 a   amar
5  5300000 6600000 m
6  6600000 8500000 á
7  8500000 18200000    r
8  18200000    19600000    ê   eu
9  19600000    21600000    u
10 21600000    22400000    â   amo
11 22400000    24000000    m
12 24000000    29700000    u
```

Figura 3.3: Excerto do ficheiro MLF com os tempos iniciais e finais de cada fone (medidos em centenas de nano-segundos).

O *software Transcriber* [23] permite transcrever e visualizar o alinhamento efetuado (figura 3.4). Para tal é necessário converter os ficheiros individuais para ficheiros compatíveis com o *Transcriber* (extensão .trs). Dado o elevado número de ficheiros que requeriam análise (5288 para cada locutor), foi realizada uma análise por estimativa, isto é, analisaram-se 200 ficheiros escolhidos de forma aleatória fazendo-se o balanço do alinhamento nestes ficheiros, concluindo-se que, em média, em 200 ficheiros, 4 apresentavam problemas de alinhamento.

[3]Trata-se de coeficientes resultantes da aplicação do sinal de fala a um banco de filtros cujas larguras de banda são definidos numa escala mel.

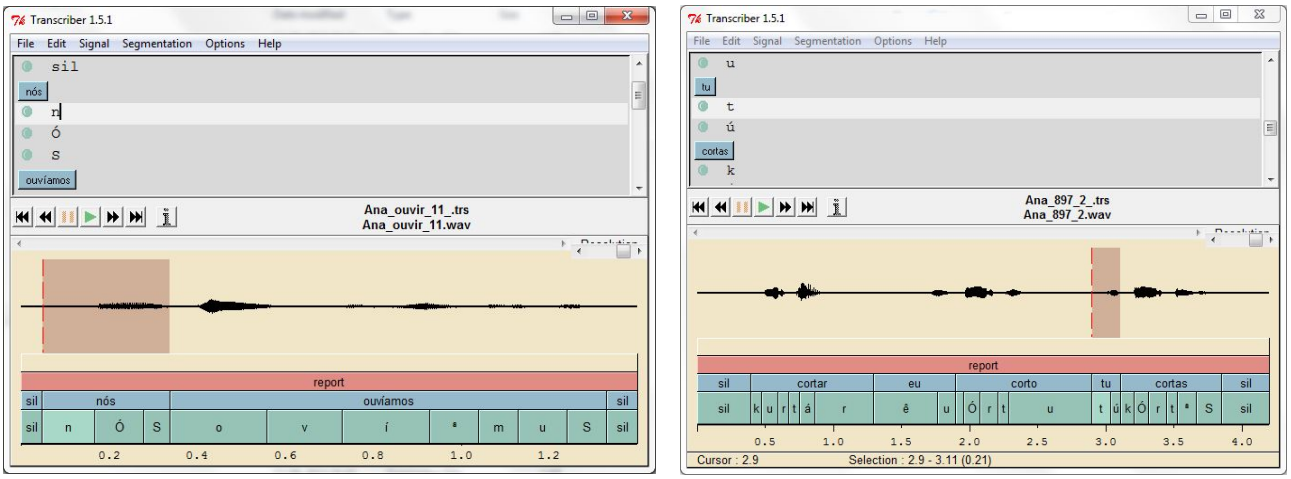


Figura 3.4: Análise do alinhamento com o *software Transcriber*

Para se criarem as etiquetas de pentafones foi utilizada uma aplicação desenvolvida numa dissertação anterior [7] que, recebendo o ficheiro MLF do alinhamento e o ficheiro de frases, cria as etiquetas de fones com contexto, seguindo o formato apresentado no anexo A.

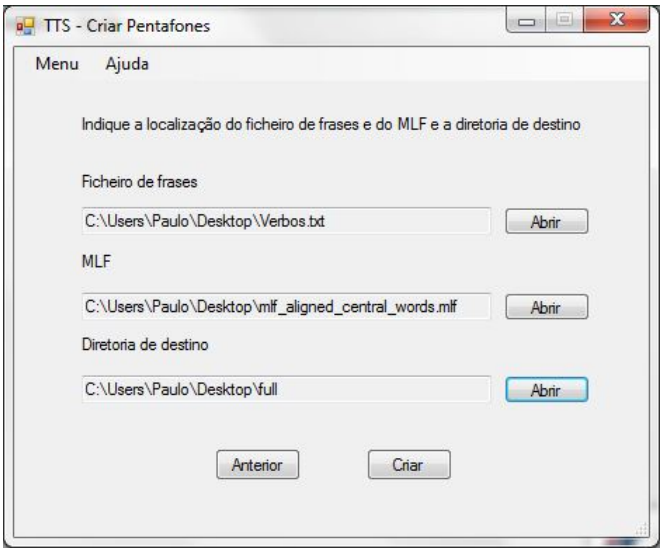


Figura 3.5: Aplicação para a criação de pentafones

3.2 Fases do treino

3.2.1 Extração de tom e parâmetros espectrais

Antes de se iniciar o treino dos modelos HMM é necessário fazer a extração dos parâmetros espectrais e do tom. O *script* de treino começa por fazer a extração dos parâmetros espectrais utilizando

as ferramentas disponíveis no SPTK [18]. Para cada ficheiro de áudio são analisadas tramas de 10 em 10ms com janelas de *Hamming* de 25ms (48000*0.025=1200 amostras) às quais é aplicada a DFT (Discrete Fourier Transform) com 2048 pontos, seguindo-se a geração de coeficientes Mel Generalized Cepstral (MGC) de ordem 35.

De seguida, é extraído o tom dos ficheiros áudio com o algoritmo de extração de tom *Snack* utilizado na *demo* HTS, que determina os excertos vozeados, não vozeados e a frequência fundamental do áudio (F0).

Por fim, são calculados os parâmetros delta e delta-delta (coeficientes de regressão linear tomando cada parâmetro em tramas anteriores e seguintes) do tom e dos parâmetros espectrais . O sistema HTS recorre a *streams* para agrupar os valores de tom e coeficientes MGC de cada ficheiro áudio, como ilustrado na figura 3.6 [14].

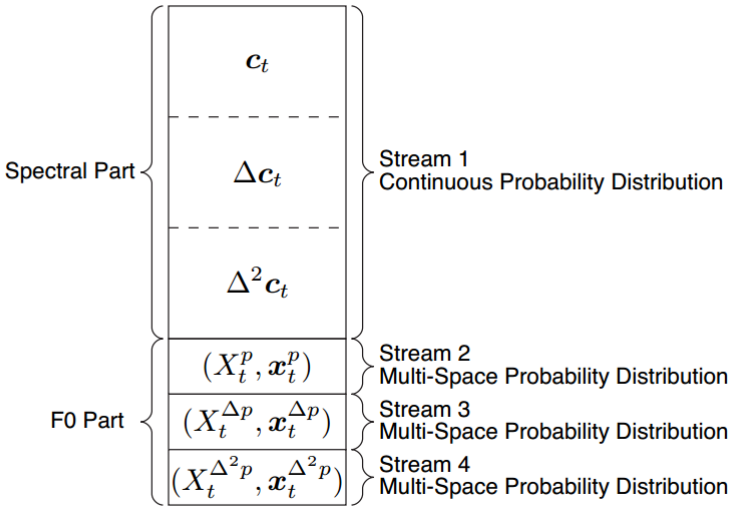


Figura 3.6: Vetor de observação. *Retirado de [14]*

3.2.2 Treino a partir de monofones

O primeiro passo nesta etapa consiste em criar protótipos de HMM, designados por "proto". Como o objetivo principal é criar uma estrutura modelo, os parâmetros não são importantes, sendo o vetor de médias preenchido a zero, o vetor de variâncias a um e a matriz de probabilidades de transição com zeros nas transições entre estados não admitidas.

De seguida é utilizada a ferramenta *HCompv* que atribui a cada modelo a variância média da base de dados. Este processo é denominado por *flat start*, pois todos os modelos são iniciados de forma igual.

O próximo passo consiste na inicialização dos fones presentes na base de dados. A partir das etiquetas de monofones são determinados todos os fones existentes e criados os seus modelos HMM.

É então realizada uma reestimação dos parâmetros recorrendo à ferramenta *HRest*, que utiliza o algoritmo de *Baum-Welch*. No final da reestimação, é, ainda, possível melhorar os modelos e é feita uma reestimação embebida com a ferramenta *HERest*.

No final da reestimação embebida é terminado o treino de monofones e iniciado o treino de fones com contexto (pentafones). Tal como nos monofones, é também necessário inicializar os modelos. A ferramenta *HHEd* faz a clonagem dos modelos HMM dos monofones para os modelos iniciais dos pentafones [22].

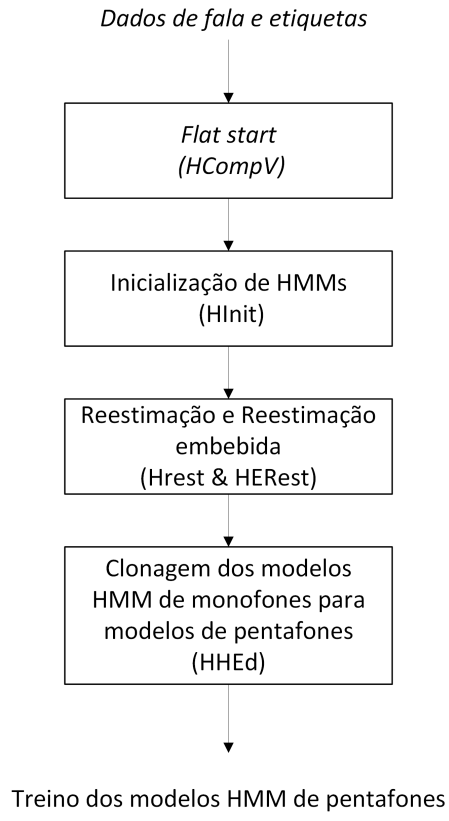


Figura 3.7: Diagrama do treino a partir de monofones. *Adaptado de [6]*

3.2.3 Treino a partir de pentafones

O teino de pentafones começa com a reestimação embebida dos parâmetros dos HMM usando a ferramenta *HERest*. Devido ao grande aumento de modelos (existem inúmeras combinações possíveis de fones para formar um pentafone) este processo não será tão preciso como no caso dos monofones.

De seguida é utilizada a ferramenta *HHEd* que constrói árvores de decisão (tom e espetro), a partir dos ficheiros de questões, onde os modelos estão agrupados em *clusters* de contexto semelhante. Este processo, usualmente chamado de *tree-based clustering*, é um passo importante pois nem sempre existe no sistema o modelo necessário na fase de síntese. Quando assim acontece, o sistema recorre às árvores de decisão e utiliza o modelo com o contexto mais semelhante ao pedido, minimizando assim a diferença ao nível da síntese.

Após a construção da árvore de decisão é feita uma reestimação dos parâmetros dos HMM de cada *cluster* (*HERest*).

O sistema HTS, neste ponto, volta a separar os modelos (*HHEd*) para melhorar o processo de treino e divisão dos modelos. É feita uma nova reestimação de parâmetros, desta vez com parâmetros partilhados (*HERest*). Finalmente, o sistema cria uma nova árvore de decisão com os modelos agrupados nos novos *clusters* (*HHEd*).

Nesta etapa é ainda estimado o modelo de duração para cada pentafone e criada uma árvore de decisão para a duração, a partir da última reestimação feita.

Na figura 3.8 está ilustrado o treino de pentafones no sistema HTS [22].

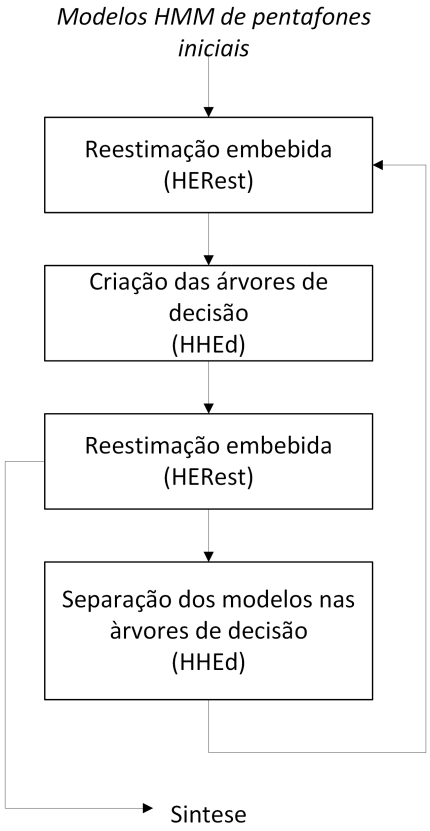


Figura 3.8: Diagrama do treino a partir de pentafones. *Adaptado de [6]*

Um último passo antes de terminar o treino é o cálculo da variância global. A utilização de componentes dinâmicas suaviza a transição entre modelos que, por vezes, pode ser de tal forma suave que se torna pouco natural. A utilização da variância global resolve este problema e aumenta assim, a qualidade da voz sintetizada, tornando-a mais natural [22] (figure 3.9).

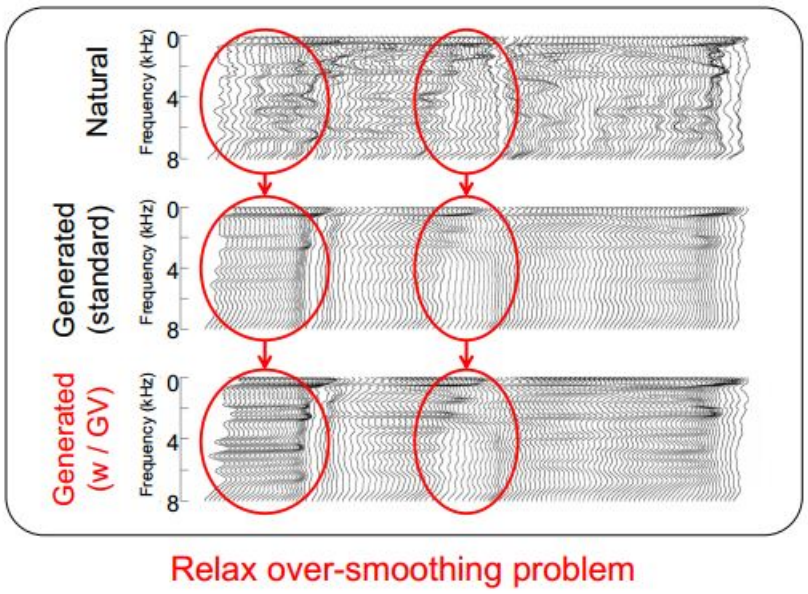


Figura 3.9: Comparação de resultados obtidos com a utilização da variância global. *Retirado de [6]*

Por fim, é feita a conversão dos ficheiros de modo a poderem ser utilizados pela ferramenta de síntese *hts_engine* [24].

3.3 Resultado do treino

Após o *script* de treino do HTS terminar (cerca de 24h para cada locutor), a voz fica definida por um conjunto de ficheiros com um tamanho total muito reduzido (cerca de 5 MB), o que elimina a barreira da capacidade de armazenamento para utilizações futuras. Estes ficheiros contêm os modelos HMM treinados, com estados partilhados pelo processo de *clustering*; as árvores de decisão para o tom, duração e parâmetros espectrais. Estas árvores de decisão auxiliam o sistema de síntese na escolha dos parâmetros do modelo de síntese (figura 1.4) mais adequado para determinado fone.

O processo de treino foi executado duas vezes, criando assim, a voz sintética de cada locutor. Estas serão as vozes utilizadas no sistema de pronunciação de verbos.

Capítulo 4

v"ErbuS - sistema online de conjugação e pronunciação de verbos

O desenvolvimento de um *website* que embebesse um sistema de conjugação e pronunciação de verbos em português europeu irá permitir a qualquer pessoa consultar a conjugação e pronunciação de um verbo de forma simples, tornando-se uma ferramenta útil, não só para utilizadores portugueses, mas também para qualquer utilizador aprendiz da língua portuguesa.

Um dos primeiros passos na criação de um *website* é a determinação do local onde o *website* será alojado. Como existem servidores dedicados a esta finalidade no laboratório, decidiu-se alojar o *site* em dois destes servidores. O servidor principal corre numa máquina com o sistema operativo *Windows 7*, que utiliza a ferramenta da *Microsoft*, Internet Information Services (IIS), para gestão do servidor *web*. Por uma questão de segurança, colocou-se o *website* num segundo servidor. Este corre numa máquina *Linux* com o sistema operativo *CentOS 6.5*, e utiliza, como servidor *web*, o *Apache* versão 2.2.15. A implementação nos dois servidores, embora tendo por base sistemas operativos diferentes, não apresenta muitas variações. A grande diferença é a linguagem utilizada no lado do servidor, *ASP* para o servidor *Windows* e *PHP* para o servidor *Linux*.

O *website* encontra-se disponível nos seguintes endereços *web*:

- `http://lsi.co.it.pt/verbos` - servidor *Windows*.
- `http://lps.co.it.pt/labfala/verbos` - servidor *Linux*.

Este site serve também de apoio *online* a um livro de divulgação científica sobre o tema da pronunciação de verbos portugueses [10].

4.1 *Layout* e descrição geral

Para se tornar uma ferramenta útil e prática de utilizar, o *site* apresenta um *layout* simples e intuitivo. Para o desenvolvimento do *site* utilizou-se o *bootstrap* [25], um *framework open-source* de Cascading Style Sheets (CSS), HyperText Markup Language (HTML) e JavaScript (JS), que fornece ferramentas com um *design* simples e moderno. A utilização deste *framework* torna também o *website* responsivo, isto é, adapta a apresentação do *website* a qualquer dispositivo, desde telemóveis a tablets ou ecrãs de computador.

Página inicial

Ao entrar no *site*, o utilizador depara-se com uma página inicial contendo o logótipo do *site*, uma caixa de entrada de texto, uma barra de navegação, uma barra de rodapé e um selecionador de linguagem (figura 4.1). Nesta página, o utilizador pode introduzir um verbo para ser conjugado através da caixa de entrada, presente na barra de navegação, ou através da caixa de entrada presente no corpo da página. A barra de navegação (comum a todas as páginas do *website*), permite regressar à página inicial, aceder à página "Acerca"ou conjugar um determinado verbo. Assim como a barra de navegação, também a barra de rodapé é comum a todas as páginas do *website* e contém uma hiperligação para o *website* do laboratório.

Figura 4.1: Página inicial do *website*

Página acerca

Nesta página é apresentada uma breve descrição do projeto, equipa técnica e referências bibliográficas.

Figura 4.2: Excerto da página "*acerca*" do *website*

Página conjugar

Após introduzir um verbo e carregar em um dos botões "*conjugar*", o utilizador é redirecionado para a página "*conjugar*", que apresenta todas as flexões e pronunciações do verbo introduzido. Nesta página, o utilizador tem a opção de ouvir a pronunciação de uma determinada flexão ou de um tempo ou forma nominal completa. É ainda fornecida a hipótese do utilizador escolher o locutor (JC ou AC), assim como o alfabeto fonológico utilizado para mostrar a pronunciação do verbo (IPA ou SAMPA).

Figura 4.3: Excerto da página *"conjugar"* do *website*

4.2 Implementação do sistema de conjugação e síntese da pronunciação de verbos

O mecanismo de conjugação e síntese da pronunciação de verbos propriamente dito, é feito com o auxilio de dois executáveis. Um executável para a conjugação do verbo, ortográfica e foneticamente (pronunciação), e outro para realizar a síntese de fala da pronunciação do verbo.

4.2.1 Conjugador de verbos

O processo de conjugar um verbo segue um conjunto de regras estabelecidas a partir da forma do infinito do verbo. Estas regras foram codificadas em *C++* seguindo um *script* em *Matlab* existente no laboratório [10]. Deste programa resulta um executável que corre no *website*. O executável recebe como parâmetro de entrada o infinitivo do verbo e começa por verificar se o mesmo se encontra numa lista (ficheiro presente no *website*) composta pelos infinitivos e respetivas transcrições fonéticas dos verbos mais utilizados na língua portuguesa. Caso o verbo esteja presente na lista, o programa guarda a sua transcrição fonética para posterior utilização na conjugação fonética do verbo; caso contrário é

utilizado o conversor de grafemas (g2p) existente no laboratório para a obter, estando o código deste processo também presente no executável.

Após obter a transcrição fonética do infinitivo do verbo, o programa começa por inicializar uma classe C++, composta por vários métodos responsáveis pela conjugação ortográfica de verbos. O infinito do verbo começa por ser analisado, definindo-se assim a *string* vogal temática (a,e,i ou o) e base a serem utilizadas. Dependendo da vogal temática, é definida a terminação do verbo (-ar,-er,-ir-or,ôr) e iniciada a função responsável pela conjugação de verbos com essa terminação. Existem quatro funções para a conjugação de verbos na classe, *conjugar_AR(), conjugar_ER(), conjugar_IR() e conjugar_OR()*. As funções *conjugar_AR(), conjugar_ER(), conjugar_IR()* apresentam um funcionamento semelhante, começando por definir uma estrutura com as terminações regulares [10] dos verbos terminados em -ar, -er ou -ir, respetivamente. De seguida é feita uma análise do verbo introduzido, verificando-se se este pertence aos verbos irregulares de cada terminação. Caso verbo introduzido seja irregular, a sua conjugação é carregada por completo, uma vez que para verbos irregulares esta foi definida manualmente em estruturas. Se o verbo introduzido não for irregular, passa-se à analise dos verbos regulares ou quase regulares, onde é por vezes necessário definir novas bases para determinados modos e tempos verbais, como por exemplo no verbo "surgir", onde na primeira pessoa do presente do indicativo, a base muda de *surg* para *surj* (surgir -> eu surjo). Nas terminações -ar e -ir, é ainda verificada a existência de ditongos na base do verbo, uma vez que na sua presença, a base fica com acento na segunda vogal do ditongo (Ex: enraizar -> eu enraízo). Nos verbos terminados em -ir é ainda necessário verificar os casos onde ocorre de mudança de vogal na base do verbo, como por exemplo no verbo dormir na primeira pessoa do presente do indicativo (eu durmo). A função *conjugar_OR()* por sua vez, inicializa a estrutura com as terminações regulares dos verbos terminados em -or e, de seguida, verifica se o verbo introduzido corresponde ao verbo pôr, onde é necessário a modificação de algumas terminações. No final das funções de conjugação, é executada a função *põe_terminações()*, que junta as bases dos tempos verbais às suas terminações correspondentes. O resultado final fica guardado numa estrutura pertencente à classe. A figura 4.4 resume o funcionamento do conjugador de verbos na forma ortográfica e as regras utilizadas no mesmo.

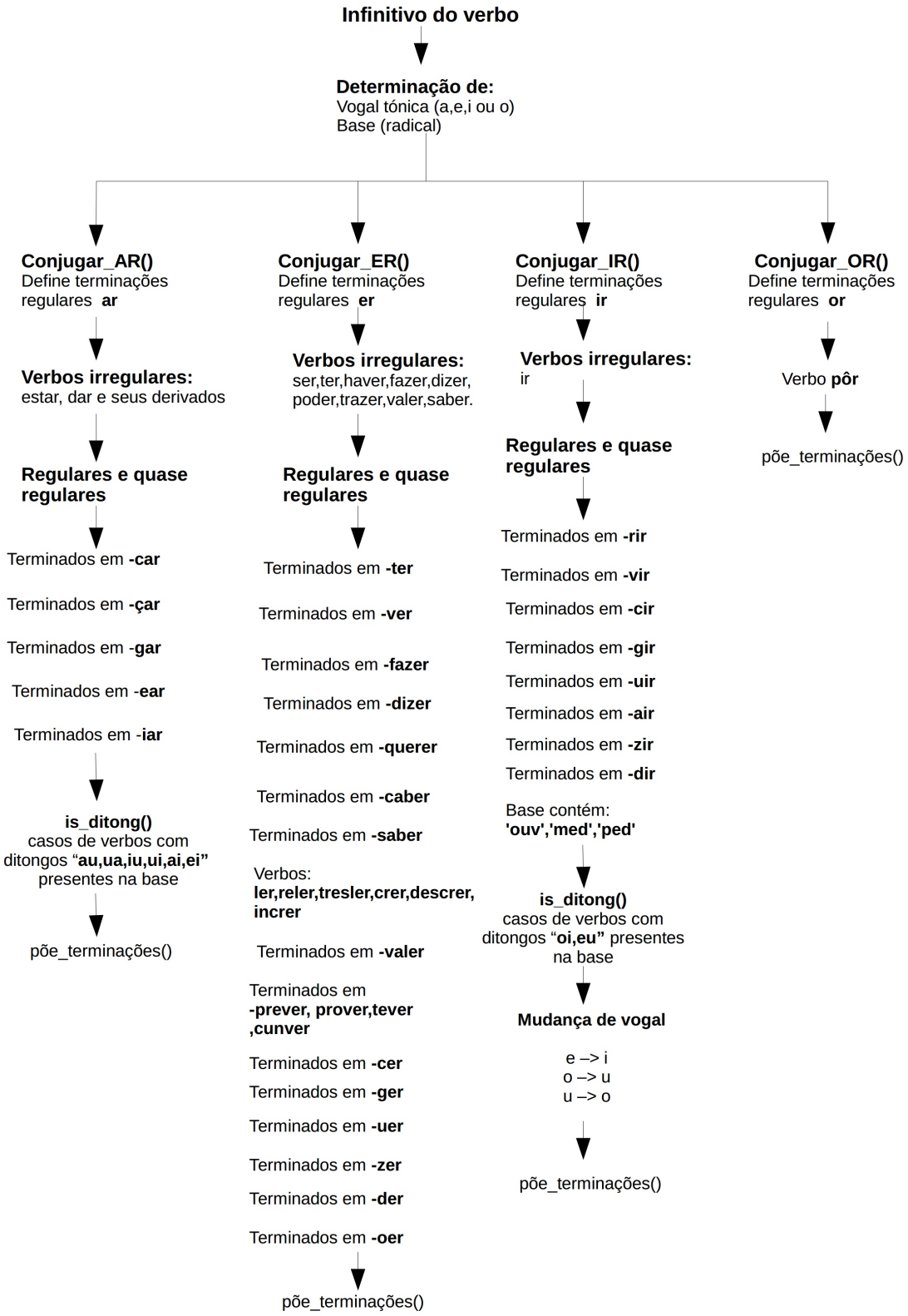


Figura 4.4: Diagrama resumido de regras do conjugador de verbos

Após conjugar ortograficamente o verbo, é inicializada uma segunda classe que procede à sua conjugação fonética (pronunciação das flexões). O mecanismo de regras utilizado nesta etapa é semelhante, com a diferença de se utilizarem transcrições fonéticas na conjugação do verbo e não as suas formas ortográficas. O alfabeto fonético utilizado nesta etapa é o SAMPA. Esta classe possui também quatro funções, responsáveis pela pronunciação dos verbos das diferentes terminações existentes. É nestas funções que se determina também o **paradigma** de pronunciação do verbo introduzido. As funções começam por definir como paradigmas os verbos *amar, viver, unir e pôr* (paradigmas dos verbos regulares na pronunciação, ver Tabela 1.3) para as terminações correspondentes, sendo este alterado quando o verbo se enquadrar numa determinada regra. Assim, para os verbos regulares que não definem nenhuma regra, os paradigmas de pronunciação são *amar, viver, unir e pôr*.

Uma vez terminada a etapa de conjugar e pronunciar o verbo, é carregado para uma *string* um ficheiro presente no servidor, que contém um protótipo da página "*conjugar*". Este protótipo possui uma tabela preenchida com parâmetros que serão substituídos pelas flexões do verbo. Os seguintes excertos de código HTML ilustram o conteúdo da tabela quando esta é carregada e após o seu processamento.

```
<td class="flex">
<p><span class="orto" id='Eu_orto2' >eu orto.PresIndic.1 </span> | <a
   href="#" style="text-decoration: none;"> <span class="pron"
   id='Eu_pron2'>"eu pron.PresIndic.1 </span> </a></p>
</td>
```

```
<td class="flex">
<p><span class="orto" id='Eu_orto2' >eu ando </span> | <a href="#"
   style="text-decoration: none;"> <span class="pron" id='Eu_pron2'>"eu
   "6~du </span> </a></p>
</td>
```

Figura 4.5: Excertos da tabela presente na página *conjugar*

Após terminado o processamento de texto do ficheiro, o resultado é enviado para a *standard output* do processo sob a forma de string.

4.2.2 Síntese de fala das flexões verbais

O executável responsável pela geração do sinal de fala correspondente à pronunciação de verbos, provém de um trabalho anterior, onde se desenvolveu um sistema *online* de síntese de fala genérico. Foram feitas algumas alterações ao programa original, de forma a ser possível escolher a voz a utilizar na síntese de fala (locutor). Este executável recebe como parâmetros de entrada os grafemas e fonemas da forma textual a ser sintetizada, assim como, o nome da pasta onde se encontram os ficheiros resultantes do treino de voz. Ao ser evocado, o programa utiliza a ferramenta hts_engine para criar uma *stream* binária, contendo o sinal de fala sintetizada, que irá ser passada para a *standard output* do processo. O facto de se utilizar uma *stream* para a *standard output* permite que o programa seja evocado inúmeras vezes em simultâneo, sem a necessidade de se criarem fisicamente ficheiros no servidor.

4.3 Análise detalhada do funcionamento do *site*

Nesta secção será feita uma análise dos mecanismos internos que ocorrem no *site*, desde a introdução de um verbo à síntese da sua pronunciação. A maioria destes processos é controlada através de JS, uma linguagem de programação dinâmica, que é executada no lado do cliente, permitindo o tratamento do conteúdo apresentado, melhorando assim a experiência do utilizador.

Como ilustrado na figura 4.1, existem duas caixas de entrada na página inicial onde é possível introduzir um verbo para ser conjugado. Nesta etapa é feita uma verificação do texto introduzido (figura 4.6), recorrendo a JS, que terá de reger-se pelos seguintes critérios:

- Possuir um comprimento maior ou igual a dois caracters (verbo ir);
- Ser composto por caracteres entre a-z ou A-Z;
- Conter uma das seguintes terminações : ar, er, ir, or, ôr.

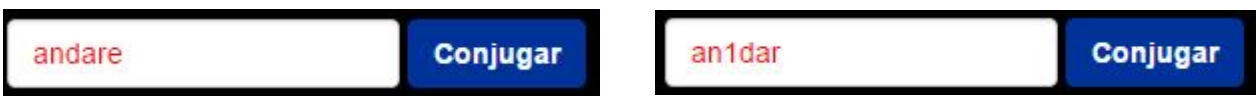


Figura 4.6: Exemplos de verificação do texto introduzido

Cumprindo os critérios supracitados, qualquer entrada é considerada válida, mesmo não tendo qualquer valor semântico.

Cada caixa de texto está inserida num *form* HTML que, após concluir a verificação do texto com sucesso, envia para a página responsável pela conjugação de verbos o texto introduzido através de um *post*. Esta página varia consoante o servidor em que se encontra o *website*.

No servidor *Linux*, o texto introduzido é enviado para uma página PHP que, com recurso à função *proc_open()* permite executar o conjugador de verbos e criar duas *pipes* para uma comunicação bidirecional com o processo. Após ser executado o programa, é passado como parâmetro de entrada o verbo a conjugar através da *pipe* dedicada para o efeito. Quando terminada a execução do programa, o seu resultado é lido através da *pipe* e gravado numa variável que é enviada diretamente para o *browser* do cliente (página conjugar).

No servidor *Windows*, devido às atualizações do IIS, por questões de segurança, deixou de ser possível executar e ter acesso ao *output* de aplicações diretamente numa página ASP. Para o efeito, são utilizados ficheiros auxiliares, designados por *handlers*. O ficheiro *handler* utilizado, do tipo .ashx, foi escrito em *C#* e funciona de forma semelhante à página PHP. Após receber o conteúdo enviado através do *form* é definido um processo e todas as suas características (programa a executar, parâmetros de entrada, etc), através da função *ProcessStartInfo()* e, iniciado com a função *Process.Start()*. O resultado do programa é, por fim, enviado para o *browser* do cliente (página conjugar).

Terminada a execução do conjugador de verbos, o utilizador encontra-se na página conjugar (figura 4.3). Nesta página, tem disponíveis as opções de reproduzir a pronunciação de um verbo, escolher o locutor utilizado na reprodução ou alterar o alfabeto fonológico, sendo que, todos estes processos são executados recorrendo a JS. Cada elemento da tabela presente na página conjugar está inserido num *span*, que possui um ID único atribuído e uma classe, como se pode verificar no excerto de código da figura 4.5. Este mecanismo de IDs permite aceder ao elemento da tabela desejado e realizar assim a operação pretendida. De forma a permitir ao utilizador interagir com as pronunciações, estas encontram-se definidas como falsos *links*, isto é, recorrendo-se à *tag* $\langle a \rangle$ definiram-se as pronunciações como *links* para o topo da página (*href="#"*). Este comportamento irá ser prevenido com recurso ao método *preventDefault()* do JS.

Quando o utilizador define o alfabeto fonológico utilizado para representar a fonética do verbo, ao alterar o valor do selecionador presente na página conjugar, é executada uma função JS que, através do método *getElementsByClassName* seleciona todos os IDs dos elementos pertencentes à classe "pron"e converte o seu conteúdo para o alfabeto escolhido (figura 4.7). De facto, ao ser carregada a página conjugar, a função *altera_ipa()* é executada, convertendo a fonética do verbo para IPA, uma vez que, como referido anteriormente, o executável "conjugador"utiliza o alfabeto SAMPA.

```
$('#dicionario').change(function () {
var e = document.getElementById("dicionario");
var strUser = e.options[e.selectedIndex].value;
if(strUser.indexOf("SAMPA")!==-1) {
altera_sampa();
}
else if ((strUser.indexOf("IPA")!==-1)) {
altera_ipa();
}
});
```

Figura 4.7: Alteração do alfabeto fonológico

Na fase de reprodução das pronunciações do verbo, optou-se por recorrer a dois sistemas diferentes, *Web Audio Application Program Interface (API)* [26] e *jPlayer* [27]. Numa primeira abordagem, utilizou-se apenas o *jplayer* para a reprodução de áudio, uma biblioteca escrita em JS que permite ultrapassar algumas das barreiras do elemento *audio* presente no HTML 5. Contudo, ao serem efetuados testes, algumas incompatibilidades entre o *jplayer*, *browsers* e dispositivos foram encontradas, nomeadamente em *iphones*, *ipads* e no *browser Safari*. De forma a colmatar este problema, substituiu-se o *jplayer* pela *Web Audio API*, uma API JS de alto nível capaz de sintetizar e processar áudio em aplicações *web*, bastante utilizada no desenvolvimento de jogos e aplicações de produção de som. O facto de ser uma API relativamente recente, leva a que alguns *browsers* ainda não tenham compatibilidade com esta. Assim, decidiu-se utilizar os dois sistemas, para cobrir o maior número de dispositivos e *browsers* possíveis. Ao ser carregada a página *conjugar*, é inicializada a utilização da *Web Audio API*, que em caso de erro, leva à utilização do JS (figura 4.8).

```
var contextClass = (window.AudioContext || window.webkitAudioContext ||
   window.mozAudioContext || window.oAudioContext ||
   window.msAudioContext);
if (contextClass) {
  var context = new contextClass();
}
else {
  is_using_web_audio=0;
}
```

Figura 4.8: Inicialização da *Web Audio API*

Uma vez que, no tratamento das locuções, se procedeu ao corte individual das flexões de todas as pessoas nos verbos paradigma, decidiu-se utilizar o áudio original na reprodução dos verbos paradigma e sintetizar os restantes. Ao clicar-se numa hiperligação, é analisado, com recurso ao método JS *event.target.id*, o ID da pronunciação (id1, figura 4.9) e, a partir deste, criado o ID que identifica a conjugação ortográfica correspondente (id2, figura 4.9) Os IDs possuem a mesma parte numérica (figura 4.5), diferindo apenas na string anterior a esta (Ex: Eu_pron2 e Eu_orto2).

```
  else if (id1.indexOf('Eu')!==-1) {
    event.preventDefault(); // previne que a posição da página se desloque
    var id2="Eu_orto"+(id1.replace( /^\D+/g, '') );
    if(is_paradigma=="0") {
      var grafemas=document.getElementById(id2).innerHTML;
      var fonemas=convert2sampa(document.getElementById(id1).innerHTML);
      var filename = ttsscript + "?grafemas=" + (grafemas.trim()) +
          "&fonemas=" + (fonemas.trim())+"_"+(locutor.trim());
    } else if(is_paradigma=="1"){
      var filename="./wav_paradigmas/"+locutor+"/"+verbo+"/"+
     verbo+"_"+(id1.replace( /^\D+/g, '') )+".wav";
    }
    playIt(filename);
  }
```

Figura 4.9: Excerto da implementação do sistema de pronunciação

Após se obterem os IDs da conjugação e da pronunciação da flexão verbal, é definida a variável *filename*. Se o verbo conjugado pertencer aos 57 verbos paradigmas da língua portuguesa, a variável *filename* irá conter o caminho físico até ao ficheiro *wav* da locução original. Aqui, o sistema de IDs implementado na tabela torna-se bastante útil, pois a sua numeração segue a mesma ordem dos ficheiros *wav*, o que permite facilmente atribuir cada ficheiro à pronunciação correspondente (Ex: ./wav_paradigmas/speaker2(Joao)/amar/amar_2.wav). No caso do verbo conjugado não ser paradigma, a variável *filename* contem o endereço para a página responsável por executar o programa de síntese, os grafemas e fonemas a sintetizar e o nome do locutor (Ex: ./tts_exec.ashx?grafemas=eu ando&fonemas="eu "6~ du_speaker2(Joao)). Novamente, o sistema de IDs implementado revela ser vantajoso, permitindo o fácil acesso aos grafemas correspondentes à pronunciação em questão. A variável *ttsscript*, que indica o nome do ficheiro que irá executar o programa, varia consoante o servidor.

Com a variavél *filename* definida, é executada a função *playIt()* (figura 4.10), que utiliza o *jplayer* ou a *Web Audio API* para reproduzir o ficheiro, dependendo do dispositivo e *browser* utilizados.

```
function playIt(url) {
  if(is_using_web_audio==1) {
    loadSoundFile(url);
  }
  else {
    if(is_paradigma==0){
      $("#gtts")
      .jPlayer("volume",0.20) // minimização da diferença de
      .jPlayer("setMedia", {wav: url }) // intensidade entre os sinais
      .jPlayer("play");
    }
    else {
      $("#gtts") // elemento <audio> do html5 presente na página
         conjugar
      .jPlayer("volume",1.0)
      .jPlayer("setMedia", {wav: url })
      .jPlayer("play");
    }}}
```

Figura 4.10: Escolha da ferramenta para reprodução de áudio

A função *playIt()* permite então reproduzir a pronunciação de uma flexão verbal, original ou sintetizada. Como existia uma diferença muito acentuada entre a intensidade dos sinais originais e sintetizados foi implementado um mecanismo que minimiza essa diferença de forma a melhorar a experiência do utilizador.

No servidor *Linux*, o ficheiro responsável pela execução do pronunciador de verbos é uma página PHP, que funciona de forma semelhante à página responsável pela execução do conjugador de verbos. Após executar o programa, a página PHP envia uma *stream* para o *browser* que efetuou o pedido contendo a fala sintetizada.

O procedimento no servidor *Windows* funciona de forma semelhante, sendo o ficheiro responsável pela execução do programa, um handler .ashx, escrito em *C#*.

A variável *filename* é então definida como caminho para o ficheiro áudio no *jplayer* ou na *Web Audio API*. Ambos os mecanismos permitem definir Uniform Resource Locators (URLs) como fonte de áudio desde que o mesmo leve a um ficheiro ou a uma *stream* de áudio. Após o processamento do pedido efetuado pelo sistema de reprodução de som, o ficheiro é, por fim, reproduzido no dispositivo do cliente, sem a necessidade de criar ficheiros temporários no servidor ou *cookies* no cliente. Esta implementação permite servir vários clientes em simultâneo, ficando limitada apenas à capacidade de processamento do servidor. A reprodução de um modo verbal completo funciona de forma semelhante. Cada botão de conjugação tem associado a si os IDs das pronunciações do modo verbal que representa, sendo possível desta forma enviar os grafemas e fonemas para o pronunciador, que devolve a fala sintetizada de todas as pessoas do modo verbal.

Dentro da página conjugar, o utilizador encontra ainda mais um *link* com o qual pode interagir. Este representa o paradigma do verbo apresentado e, quando clicado, leva a que seja efetuado um *post* para a página que executa o programa de conjugação de verbos. Como se trata de um *link* normal, não se pode recorrer a um *form* normal de HTML, pelo que se utilizou JS para fazer o *post*.

Em qualquer altura, o utilizador pode alterar o locutor que pretende para as reproduções das pronunciações do verbo, recorrendo ao selecionador existente na página. O valor presente no selecionador irá ser o valor enviado ao sintetizador, que irá utilizar os ficheiros do locutor em questão. O mesmo acontece para a pronunciação de verbos paradigma, em que o valor determina o nome da pasta em que a locução original se encontra.

Encontra-se então finalizada, a análise dos processos internos mais importantes que ocorrem no *website* aquando da sua utilização.

Capítulo 5

Análise de resultados

Concluído o objetivo proposto para esta dissertação, é agora necessário realizar uma análise dos resultados obtidos. Devido ao facto do produto final desta dissertação ser um *website*, a sua análise irá ter dois focos principais: imagem e usabilidade do *website* e qualidade do sintetizador. A avaliação dos dois parâmetros mencionados irá ser feita com recurso a um questionário, de forma a recolher dados que permitam uma avaliação subjetiva.

5.1 Análise subjetiva

Para avaliar o resultado final foi realizada uma análise subjetiva. Esta teve por base um questionário que foi respondido por um grupo de avaliadores externo ao projeto, composto por vinte e três elementos. O questionário continha perguntas relacionadas com o *website* e sintetizador de fala. Na primeira página eram apresentadas duas perguntas relacionadas com o aspeto gráfico e funcionalidade do *website*. As perguntas e os seus resultados obtidos encontram-se na figura 5.1.

Para a primeira questão a média foi de 4.57 e para a segunda de 4.39.

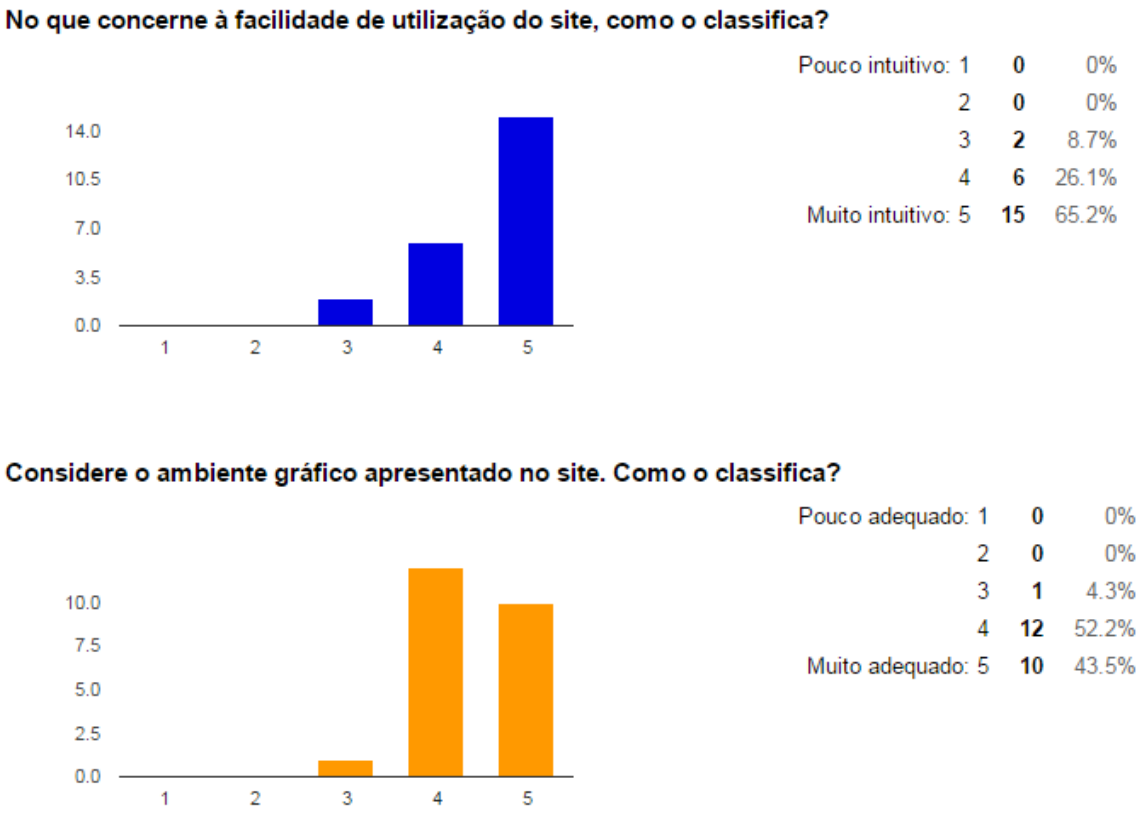


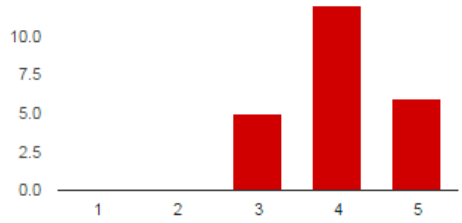
Figura 5.1: Resultados relativos à avaliação do *website*

De seguida foi realizada a avaliação da naturalidade e semelhança do sintetizador à voz original. Aqui os avaliadores dispunham de dois pares de vídeos constituídos pela pronunciação original e sintetizada de uma determinada flexão para cada locutor. Após analisarem os vídeos, eram colocadas duas questões relativas aos mesmos (figura 5.2 e 5.3).

Na avaliação da locutora AC a média das respostas foi de 4.04 e 3.91 para a primeira e segunda questão, respetivamente.

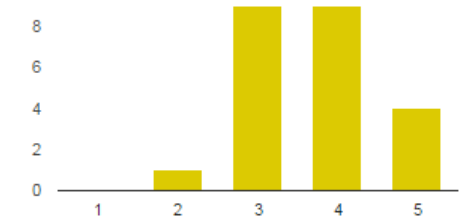
Para o locutor JC, a média da primeira questão foi de 4.35 e 4.39 para a segunda.

Após ouvir os ficheiros originais e sintetizados, como classifica a semelhança entre as vozes sintetizadas e a originais?



Pouco semelhante:	1	0	0%
	2	0	0%
	3	5	21.7%
	4	12	52.2%
Muito semelhante:	5	6	26.1%

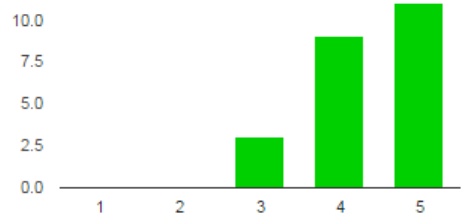
Como classifica a voz sintetizada quanto à sua naturalidade?



Pouco natural:	1	0	0%
	2	1	4.3%
	3	9	39.1%
	4	9	39.1%
Muito natural:	5	4	17.4%

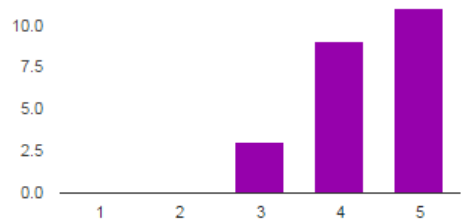
Figura 5.2: Resultados relativos à voz sintetizada AC

Após ouvir os ficheiros originais e sintetizados, como classifica a semelhança entre as vozes sintetizadas e a originais?



Pouco semelhante:	1	0	0%
	2	0	0%
	3	3	13%
	4	9	39.1%
Muito semelhante:	5	11	47.8%

Como classifica a voz sintetizada quanto à sua naturalidade?



Pouco natural:	1	0	0%
	2	0	0%
	3	3	13%
	4	9	39.1%
Muito natural:	5	11	47.8%

Figura 5.3: Resultados relativos à voz sintetizada JC

Por último, foi realizada uma avaliação da perceção da fala. Para tal, foram colocados oito vídeos no questionário, onde os avaliadores, depois de ouvirem cada um deles, teriam de transcrever a frase. As oito frases (quatro de cada locutor) presentes nos vídeos eram as seguintes:

AC

1. que nós transpareçamos
2. ele abrilhantara
3. para eu avaliar
4. vós permitiríeis

JC

1. se nós contornássemos
2. eu recomporia
3. quando eu institucionalizar
4. nós saltávamos

Os resultados das transcrições das frases sintetizadas estão presentes na tabela 5.1.

	Número de transcrições corretas	Número de transcrições incorretas	Percentagem de frases corretas
Frases		**AC**	
1	23	0	100%
2	20	3	86.95%
3	23	0	100%
4	23	0	100%
TOTAL	89	3	96.73%
Frases		**JC**	
1	19	4	82.6%
2	18	5	78.26%
3	23	0	100%
4	23	0	100%
TOTAL	83	9	90.22%

Tabela 5.1: Resultados da avaliação da perceção da fala sintetizada

Após analisar os resultados do questionário, pode concluir-se que, de um modo geral, o projeto teve uma avaliação positiva. O *website* é considerado atrativo e intuitivo. Em relação ao sintetizador, verificou-se que, para a locutora AC, os dados diferem dos do locutor JC, sendo a maior diferença na naturalidade da voz. Este facto poderá estar relacionado com a extração de tom que, não sendo executado com os limites adequados, poderá levar a uma síntese não tão natural em certas palavras.

Capítulo 6

Conclusão

O objetivo definido para esta dissertação, que consistia no desenvolvimento de um sistema *online* de conjugação e pronunciação de verbos para o português europeu, foi cumprido com sucesso. O *website* desenvolvido possibilita a qualquer utilizador o fácil e rápido acesso a uma determinada conjugação ou pronunciação de um dado verbo. Esta ferramenta pode tornar-se bastante útil para aprendizes da língua portuguesa, bem como para qualquer falante nativo desta, permitindo esclarecer dúvidas de cariz ortográfico ou fonético. O facto de ser possível a reprodução de uma dada flexão verbal através de síntese de voz, destaca este projeto das outras ferramentas existentes na internet. A voz sintetizada mostrou ter uma qualidade muito boa, tanto para o locutor masculino JC, como para a locutora feminina AC.

Ao longo do desenvolvimento deste projeto foram adquiridos e melhorados vários conhecimentos técnicos, uma vez que, o projeto envolveu a utilização constante de variadas linguagens de programação, nomeadamente *C++, Matlab*, JS, PHP, ASP e HTML. Foi também necessário a utilização de dois sistemas operativos distintos (*Windows* e *Linux*) e a utilização de ferramentas para gestão de *websites*, como o IIS.

Embora o desenvolvimento do *website* tenha tido em conta as várias tipologias de dispositivos existentes hoje em dia e apresente um funcionamento normal nos mesmos, a adaptação deste projeto a dispositivos móveis por parte de uma aplicação (*Windows phone*, *Android*, *iOS*) seria um ponto forte a implementar. Em relação à qualidade final da fala sintetizada, esta está relacionada diretamente com a base de dados utilizada no treino. Assim, um aumento das locuções que constituem a base de dados, acrescentando frases foneticamente ricas e diferentes de flexões verbais, levaria a um aumento da qualidade final da fala sintetizada.

Bibliografia

[1] Camões - instituto da cooperação e da língua [Online]. http://www.instituto-camoes.pt/, Agosto 2015.

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

[3] Speech synthesis [Online]. https://en.wikipedia.org/wiki/Speech_synthesis, Agosto 2015.

[4] K. Tokuda, Y. Nankaku, H. Zen T. Toda, J. Yamagishi, and K. Oura. Speech Synthesis Based on Hidden Markov Models. *Proceedings of the IEEE*, 101(5):1234–1252, May 2013.

[5] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, T. Toda, A.W.Black, T. Nose, and K. Oura. HMM-based speech synthesis system (HTS). http://hts.sp.nitech.ac.jp/.

[6] HTS Slides. Hmm-based speech synthesis system (hts). http://hts.sp.nitech.ac.jp/?Download, 2010.

[7] Tiago Ferreira. *Sistema Online de Síntese de Fala em Português*. PhD thesis, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2014.

[8] João Gomes. *Treino de Modelos para um Sistema de Síntese de Fala em Português*. PhD thesis, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2011.

[9] IPA [Online]. https://www.internationalphoneticassociation.org/, Agosto 2015.

[10] A. Veiga, S. Candeias, and F. Perdigão. *Pronunciação de Verbos Portugueses - Guia Prático*. Lidel, Lisboa, Portugal, 2015.

[11] SAMPA [Online]. http://www.phon.ucl.ac.uk/home/sampa/portug.htm, Agosto 2015.

[12] C. Cunha and L. Cintra. *Nova Gramática do Português Contemporâneo*. Nova Fronteira S.A., Rio de Janeiro, Brasil, 3 edition, 2001.

[13] Flip gramática [Online]. http://www.flip.pt/FLiP-On-line/Gramatica.aspx, Agosto 2015.

[14] J. Yamagishi. An Introduction to HMM-Based Speech Synthesis. October 2006.

[15] Cambridge University Engineering Department. Hidden Markov Model Toolkit (HTK). http://htk.eng.cam.ac.uk/.

[16] S. Imai, K. Sumita, and C. Furuichi. Mel log spectrum approximation (MLSA) filter for speech synthesis. *Electronics and Communications in Japan*, 66:10–18, 1983.

[17] Mel scale [Online]. https\protect\kern+.2222em\relax//en.wikipedia.org/wiki/Mel_scale, Agosto 2015.

[18] Sptk [Online]. http://sp-tk.sourceforge.net/, Junho 2015.

[19] Sox [Online]. http://sox.sourceforge.net/, Junho 2015.

[20] Audacity [Online]. http://audacityteam.org/, Junho 2015.

[21] Instituto de Telecomunicações Laboratório de Processamento de Sinal. Grafone. http://lsi.co.it.pt/spl/g2p/.

[22] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. "the hidden markov model toolkit (htk) version 3.4"[Online]. http://htk.eng.cam.ac.uk/, 2006.

[23] Transcriber [Online]. http://trans.sourceforge.net/en/presentation.php, Junho 2015.

[24] hts_engine api [Online]. http://hts-engine.sourceforge.net/, Junho 2015.

[25] bootstrap [Online]. http://getbootstrap.com/, Junho 2015.

[26] B. Smus. *Web Audio API: Advanced Sound for Games and Interactive Apps*. Nova Fronteira S.A., United States of America, 1 edition, 2013.

[27] jplayer [Online]. http://jplayer.org/, Junho 2015.

Anexo A

An example of contextdependent label format for HMMbased speech synthesis in European Portuguese

Tiago Ferreira
June 16, 2014

$m_1\hat{}m_2-m_3+m_4=m_5$ /M2:$m_6\_m_7$
/S1:s_1_@s_2s_3_@s_4+s_5_@s_6 /S2:$s_7\_s_8$ /S3:$s_9\_s_{10}$ /S4:$s_{11}\_s_{12}$ /S5:$s_{13}\_s_{14}$ /S6:s_{15}
/W1:w_1_#w_2-w_3_#w_4+w_5_#w_6 /W2:$w_7\_w_8$ /W3:$w_9\_w_{10}$ /W4:$w_{11}\_w_{12}$ /W5:w_{13}
/W6:$w_{14}\_w_{15}$ /W7:$w_{16}\_w_{17}$
/P1:p_1_!p_2p_3_!p_4+p_5_!p_6 /P2:$p_7\_p_8$ /P3:p_9
/U:u_1_u_2_&u_3

m_1	the phoneme identity before the previous phoneme
m_2	the previous phoneme identity
m_3	the current phoneme identity
m_4	the next phoneme identity
m_5	the phoneme after the next phoneme identity
m_6	position of the current phoneme identity in the current syllable (forward)
m_7	position of the current phoneme identity in the current syllable (backward)
s_1	whether the previous syllable stressed or not (0: not stressed, 1: stressed)
s_2	the number of phonemes in the previous syllable
s_3	whether the current syllable stressed or not (0: not stressed, 1: stressed)
s_4	the number of phonemes in the current syllable
s_5	whether the next syllable stressed or not (0: not stressed, 1: stressed)
s_6	the number of phonemes in the next syllable
s_7	position of the current syllable in the current word (forward)
s_8	position of the current syllable in the current word (backward)
s_9	position of the current syllable in the current phrase (forward)
s_{10}	position of the current syllable in the current phrase (backward)
s_{11}	the number of stressed syllables before the current syllable in the current phrase
s_{12}	the number of stressed syllables after the current syllable in the current phrase
s_{13}	the number of syllables, counting from the previous stressed syllable to the current syllable in this utterance
s_{14}	the number of syllables, counting from the current syllable to the next stressed syllable in this utterance
s_{15}	name of the vowel of the current syllable
w_1	part-of-speech classification of the previous word
w_2	the number of syllables in the previous word
w_3	part-of-speech of classification of the current word
w_4	the number of syllables in the current word
w_5	part-of-speech classification of the next word
w_6	the number of syllables in the next word
w_7	position of the current word in the current phrase (forward)

w_8	position of the current word in the current phrase (backward)
w_9	the number of content words before the current word in the current phrase
w_{10}	the number of content words after the current word in the current phrase
w_{11}	the number of words counting from the previous content word to the current word in this utterance
w_{12}	the number of words counting from the current word to the next content word in this utterance
w_{13}	punctuation after the previous word
w_{14}	punctuation before the current word
w_{15}	punctuation after the current word
w_{16}	punctuation after the next word
w_{17}	punctuation after the second next word
p_1	the number of syllables in the previous phrase
p_2	the number of words in the previous phrase
p_3	the number of syllables in the current phrase
p_4	the number of words in the current phrase
p_5	the number of syllables in the next phrase
p_6	the number of words in the next phrase
p_7	position of the current phrase in this utterance (forward)
p_8	position of the current phrase in this utterance (backward)
p_9	ending punctuation of the current phrase
u_1	the number of syllables in this utterance
u_2	the number of words in this utterance
u_3	the number of phrases in this utterance