FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Departamento de Engenharia Electrotécnica e de Computadores

RECONHECIMENTO AUTOMÁTICO DE FALA COM PROCESSAMENTO SIMULTÂNEO DE CARACTERÍSTICAS ACÚSTICAS E VISUAIS

ANTÓNIO AFONSO DE ABREU E MOURA

Licenciado em Engenharia Electrotécnica e de Computadores pela
Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos requisitos para obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores

Dissertação realizada sob a supervisão de
Vítor Manuel Martins Cicouro de Pêra, Professor Auxiliar, e
Diamantino Rui da Silva Freitas, Professor Associado do,
do Departamento de Engenharia Electrotécnica e Computadores
da Faculdade de Engenharia da Universidade do Porto

PORTO, MAIO DE 2005

RECONHECIMENTO AUTOMÁTICO DE FALA COM PROCESSAMENTO SIMULTÂNEO DE CARACTERÍSTICAS ACÚSTICAS E VISUAIS

ANTÓNIO AFONSO DE ABREU E MOURA

Dissertação submetida para satisfação parcial dos requisitos para obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores

À Sílvia, à minha irmã e ao meu pai,
por todo o apoio e compreensão
ao longo dos últimos anos

RESUMO

Na presente dissertação é apresentada uma abordagem ao estudo e desenvolvimento de um reconhecedor automático de fala, para o Português Europeu, com processamento simultâneo de características acústicas e visuais. O objectivo deste trabalho consiste em dar mais um passo no reconhecimento automático de fala para o Português Europeu, utilizando o processamento das características visuais de modo a conferir maior robustez aos sistemas existentes.

A primeira abordagem consiste num trabalho de pesquisa, na área de reconhecimento automático de fala, com a finalidade de conhecer as técnicas existentes. Nesta fase foi ainda definida a aplicação de reconhecimento de fala e seleccionada a metodologia mais adequada.

Depois de definida a aplicação foi necessário criar e desenvolver uma base de dados de suporte ao estudo, devido à especificidade da aplicação e à falta de recursos nesta área para o Português Europeu. Neste desenvolvimento houve a necessidade de criar uma base de dados de raiz. Foi necessário fazer a segmentação e etiquetagem da base de dados e desenvolver um algoritmo para extracção das características acústicas e visuais. No que respeita às características acústicas a metodologia seguida foi a extracção dos coeficientes Mel-Cepstrais (MFCC). Para extracção das características visuais foi desenvolvido um algoritmo para encontrar a ROI (Region of Interest) e de seguida aplicada a essa região a DCT2 (*Discrete Coussine Transform*).

Depois de extraídos os coeficientes visuais, passou-se ao objectivo principal desta dissertação, o desenvolvimento do sistema de reconhecimento. Numa primeira fase visou-se a construção do sistema-base de reconhecimento. Foi seguida uma abordagem inicial single- stream, desenvolvendo-se o sistema de reconhecimento para características acústicas e visuais separadamente. De seguida foi desenvolvido um sistema multi-stream baseado nos sistemas single-stream.

O culminar deste trabalho consiste no desenvolvimento do sistema multi-stream baseado nas características acústicas e visuais. Este desenvolvimento proporcionou melhoramento da taxa de acerto do sistema base nomeadamente na presença de condições de reconhecimento adversas (a nível acústico).

ABSTRACT

In the dissertation, it is presented one view to the study and development of a automatic speech recognizer (European Portuguese), with simultaneous processing of acoustic and visual characteristics. The objective of this work is to give one more step in the automatic speech recognition to the European Portuguese, using the processing of visual characteristics in a way to increase the actual systems.

The first approach consists of a research work in the field of automatic speech recognition, with the purpose of knowing the existent tecniques. At this step was defined the application of speech recognition and the most adequate methodology was selected.

After defining the application, it was necessary to creat and develops a support database, because of the specificity of the subject and the lack of resources in this area to the European Portuguese. In this development there was the need to creat a database from scratch. It was necessary to make the segmentation and tagging of the database and to develop an algorithm for extraction of the acoustic and visual characteristics. Regarding to the acoustic characteristic, the methodology followed was the extraction of mel-cepstrals coefficients (MFCC). For the extraction of the characteristics, it was developed an algorithm to find a ROI (*Region of Interest*) followed by the application of a DCT2 (*Discrete Cosine Transform*).

After having extracted the visual coefficients, the principal objective of the dissertation was in reach: The development of the automatic speech recognition system. On a first phase, the base-system was developed following a single-stream approach, developing the recognition system for acoustic characteristics and for visual characteristics in a separate way. Based on the single-streams, a multi-stream system was developed.

The final step of this work consists in the development multi-stream system based on acoustic and visual characteristics. This development targets the increase of the hitrate of the system, especially in the presence of adverse acoustic recognition conditions.

AGRADECIMENTOS

A concepção deste trabalho não teria sido possível sem a colaboração de diversas pessoas que contribuindo de uma forma positiva, criaram condições óptimas para o seu bom desenvolvimento.

Aos professores Vítor Pêra e Diamantino Freitas, pela sua dedicação, incentivos e disponibilidade e por me terem dado a oportunidade de realizar este trabalho.

A todos os colegas e companheiros de laboratório que durante a realização do trabalho se mostraram críticos relativamente a muitos dos aspectos associados ao mesmo.

Ao Pedro, e à sua família, por se terem disponibilizado sem reservas para recolhas da base de dados sem a qual este trabalho não se poderia ter realizado dentro dos objectivos pretendidos.

À Dora e à Maria Fernada pelo importante contributo que deram a este trabalho.

Ao meu pai e à minha irmã por todo o apoio e confiança que sempre me transmitiram.

Por último mas não menos importante, quero deixar uma palavra de carinho para a Sílvia pelo seu amor e compreensão.

O desenvolvimento deste trabalho foi financiado pelo Programa de Desenvolvimento Educativo para Portugal (PRODEP III). Esta dissertação foi realizada nas instalações do Laboratório de Processamento da Fala, Electrónica Sinais e Instrumentação da FEUP com a duração de 8 meses.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

Introdução

Neste capítulo é realizada uma introdução às noções básicas relacionadas com o reconhecimento automático da fala. Inicialmente e de modo sucinto são apresentados os processos de produção, transmissão e recepção da fala. Seguidamente é realizada uma análise sobre o problema do reconhecimento da fala, a complexidade e dificuldade, nomeadamente no que diz respeito aos sistemas desenvolvidos para aplicações de fala contínua. Posteriormente é apresentada uma descrição da evolução dos sistemas de reconhecimento de fala, e estado actual, focando principalmente os sistemas de reconhecimento com processamento de características acústicas e visuais.

1.1 A Comunicação Oral

Desde que nos conhecemos, comunicamos geralmente todos os dias e aceitamos esse processo com naturalidade. Apesar da comunicação entre as pessoas ser um processo simples e natural, a mensagem que pretendemos transmitir no sinal[1] sonoro que produzimos, assim como, a sua

[1] É o agrupamento de vários sinais que podem ser observados na comunicação.

interpretação, pelo ouvinte da mensagem por nós transmitida, podem constituir um conjunto de processos bastante complexos.

O conteúdo da mensagem observado pelo ouvinte depende do contexto da comunicação. Em comunicações em que as pessoas têm contacto visual[1], juntamente com o sinal acústico são observadas alterações físicas provocadas na formulação da mensagem, que podem ser representadas num sinal visual. Nas comunicações em que os elementos que estão em contacto não estão no campo visual uma das outras[2], a recuperação do conteúdo será realizada com maior dificuldade, apenas com o sinal acústico.

1.1.1 Modelo da Comunicação através da fala

O processo de produção da fala começa quando o orador[3] formula conceptualmente a mensagem, que pretende transmitir através de fala, ao seu ouvinte[4]. Essa mensagem, restringida pelas regras da linguagem utilizada, é transformada numa mensagem linguística.

Deste modo, a mensagem é transformada num conjunto de sons, assim como num conjunto de movimentos faciais, que correspondem às palavras da mesma. Para produzir esses sons, é desencadeada uma série de acções neuromusculares que irão gerar a forma apropriada do tracto vocal ou então provocar a vibração das cordas vocais (secção 1.2.1). Essas acções neuromusculares levam a alterações dinâmicas da forma da face, principalmente associadas à zona da boca, fundamental no processo de modelação dos sons que se pretendem transmitir.

O sinal de fala, depois de ser gerado, propaga-se pelo ar do orador até ao ouvinte. O ouvinte dá início ao processo de audição e percepção da fala. Se o tempo decorrido não for elevado, pode contar-se com uma contribuição redundante entre a imagem e o som. O sinal acústico é processado pela cóclea e nesta pela membrana basilar, no ouvido interno que realiza uma forma de análise espectral do sinal. O espectro do sinal de vibração gerado pela referida membrana é convertido em sinais de activação no nervo auditivo, através de um processo de transdução neuronal. Este processo pode assemelhar-se a um processo de extracção de características. Da

[1] Comunicação presencial (a mais usual), video-conferência, etc.
[2] Comunicação telefónica, radio-frequência, etc.
[3] Na literatura podem encontradas designações como falante ou informante. Apesar de se poder deiferenciar entre cada tipo, sempre que nesta dissertação se utilizar uma das designações referidas, o objectivo é indentificar o emissor da menssagem.
[4] Receptor da menssagem

actividade neuronal, envolvendo o nervo auditivo e o córtex cerebral, surge neste um código da linguagem. É no córtex cerebral que se dá a interpretação e compreensão da mensagem recebida.

Simultâneamente com o processamento do sinal acústico, sempre que o ouvinte consegue visualizar a cara do orador, as alterações provocadas nesta durante a produção da mensagem são recolhidas e processadas pelo cérebro, ajudando na interpretação da mensagem. Apesar de nem sempre ser relevante no processo de comunicação oral, a componente visual pode ajudar em comunicações com condições acústicas desfavoráveis[1]. Uma vantagem do sinal visual é a facilidade com que se consegue transmitir e interpretar as emoções[2] (as emoções podem provocar uma grande variabilidade no sinal acústico levando a uma diminuição do desempenho de sistemas de reconhecimento se não forem robustos contra estas variações), processo nem sempre simples através do sinal acústico.

Na Figura 1.1 é apresentado um modelo esquemático do processo de comunicação humana através da fala.

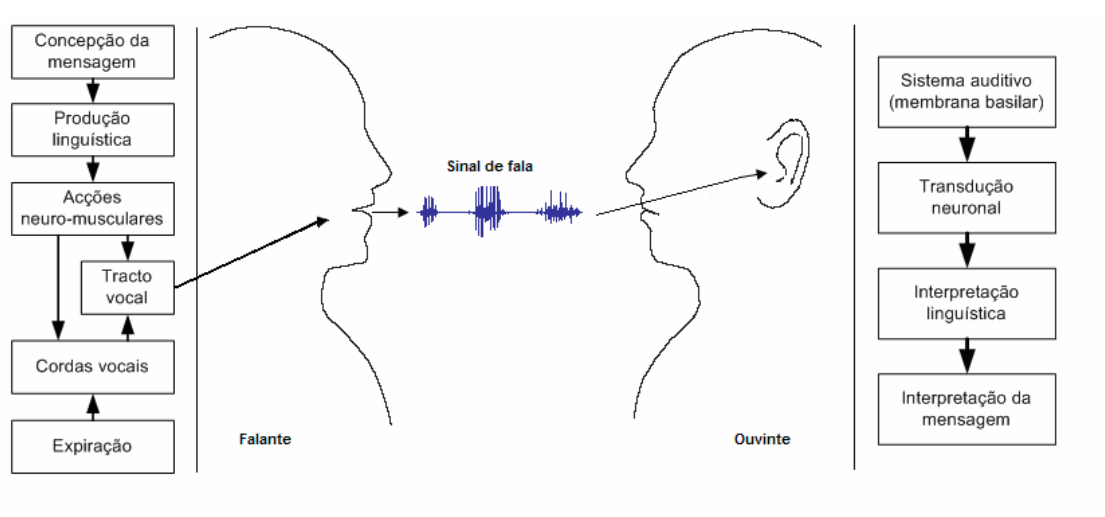


Figura 1.1 - Modelo esquemático da comunicação humana através da fala. (Figura adaptada de [1]).

[1] Especialmente para pessoas com capacidade de leitura labial
[2] Alegria, tristeza, ansiedade, nervosismo, etc.

1.2 A fala

Os sinais de fala, presentes no processo de comunicação oral, são compostos por uma sequência de sons[1] e alterações da face[2], regulados pelas regras da língua e pelas características do orador. Para processar sinais de fala[3] é necessário perceber correctamente o mecanismo da sua produção. Vai ser realizada uma primeira abordagem mais concentrada na produção do sinal acústico, seguida por uma abordagem, na secção seguinte, à produção do sinal visual.

1.2.1 A Produção da fala e o Aparelho Fonador

A produção da fala depende não só dos órgãos específicos, mas também do sistema de respiração, uma vez que os sons emitidos resultam quase sempre da acção desses órgãos sobre a corrente de ar oriunda dos pulmões. O ar que expiramos, quando sujeito a variações de pressão e volume pelos órgãos do sistema[4] de produção da fala, cria as ondas sonoras que a caracterizam. A inspiração é normalmente associada a períodos de silêncio. O aparelho fonador humano é apresentado na Figura 1.2 e é constituído por:

- Órgãos respiratórios que fornecem a corrente de ar (garantem o fluxo de ar): pulmões, brônquios e traqueia;
- Órgão que constitui a principal fonte de conteúdo sonoro utilizado na fala (que se comporta como obstáculo à passagem de ar e que é responsável pelo tipo de excitação fornecido às cavidades superiores): laringe, onde se encontram as cordas vocais;
- Órgãos que funcionam como caixas de ressonância: cavidades supra laríngeas (faringe, boca e fossas nasais).

[1] Originando um fenómeno acústico associado
[2] Originando um fenómeno visual associado
[3] Reconhecer ou sintetizar fala
[4] O aparelho fonador

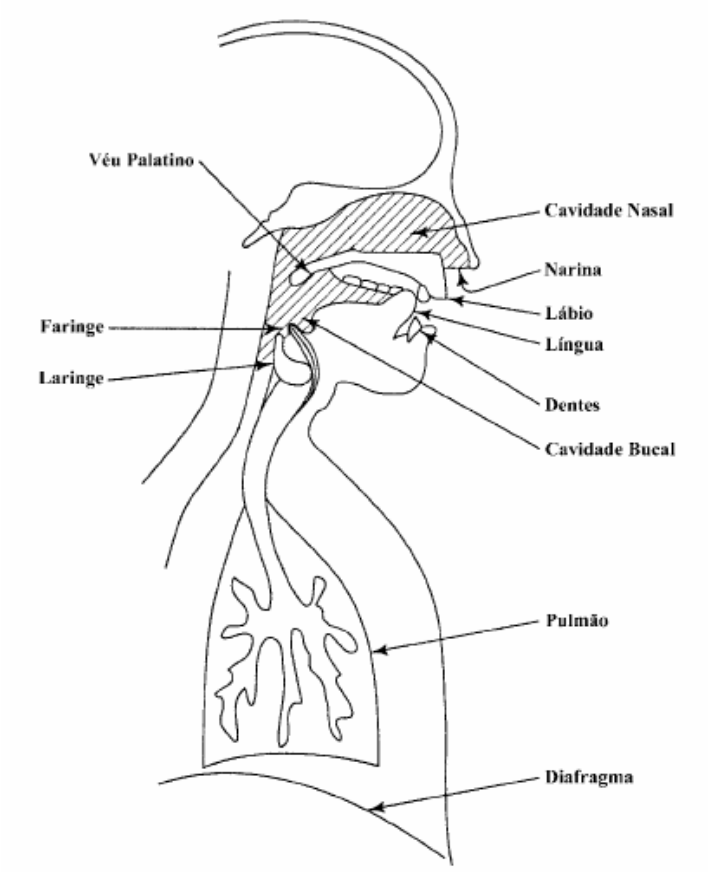


Figura 1.2 - O Aparelho Fonador humano. (Figura adaptada de [2]).

A faringe e a cavidade oral são normalmente agrupadas numa unidade designada por tracto vocal. A cavidade nasal pode também ser designada por tracto nasal.

Os componentes anatómicos mais finos, ou seja, as cordas vocais, o véu palatino, a língua, os dentes e os lábios, são designados por articuladores. Esta designação advém do facto de serem a posição e a dinâmica do movimento destes componentes a modelar os sons, determinando a forma do tracto vocal, em cada instante.

O percurso da voz é idêntico ao da respiração. Durante a inspiração o volume dos pulmões aumenta, provocando dessa forma a diminuição da pressão de ar e a consequente entrada do mesmo para os pulmões. Quando a inspiração termina, a pressão do ar no interior dos pulmões volta a ser semelhante à atmosférica, terminando o fluxo de ar. Logo de seguida ocorre o processo inverso, ou seja, diminui o volume dos pulmões, aumentando a pressão do ar relativamente à pressão atmosférica, originando um fluxo de ar dos pulmões para o exterior. É nesta fase, em que o ar enviado pelos pulmões atravessa a traqueia e a laringe em direcção à

faringe, que se dá a ligação às cavidades nasal e bucal. Os órgãos de fonação, dispostos ao longo do percurso atravessado pelo fluxo de ar, são responsáveis durante a expiração por uma maior pressão sobre o ar circulante, tornando audível esse fluxo.

A laringe é um dos órgãos mais importantes na fonação. É constituída por cartilagens, e encontra-se suspensa por membranas e músculos, podendo mover-se verticalmente e na direcção antero-posterior e assim alterar ligeiramente o volume das cavidades supraglotais e consequentemente as condições acústicas do ar nessas cavidades. No seu interior situam-se as cordas vocais, formadas por pregas musculares localizadas nas paredes superiores da laringe. A glote representa a abertura entre essas duas pregas, que constituem o primeiro obstáculo ao fluxo de ar oriundo dos pulmões.

Durante a respiração normal, a glote encontra-se sempre aberta[1], permitindo dessa forma a livre circulação do ar. Durante a fonação, as cordas vocais vibram, abrindo e fechando rapidamente a passagem ao fluxo de ar vindo dos pulmões. A junção das cordas vocais, conduz ao aumento da pressão subglotal até a um ponto que obriga ao afastamento entre as cordas. O afastamento permite a libertação de ar e a consequente diminuição da pressão subglotal, levando desse modo a nova aproximação. A repetição cíclica deste processo produz a vibração das cordas vocais durante a fonação.

1.2.2 O Tracto Vocal

O tracto vocal, durante a produção do som, é o responsável por colocar barreiras de tipos muito variados ao fluxo de ar expelido pelos pulmões, sendo pois lógico considerar que se comporta, nesta fase, como um tubo[2] de ressonância.

As ressonâncias do tracto vocal podem ser modeladas pela cavidade bucal ou pela cavidade faríngea. No modelo da cavidade bucal o objectivo é representar o tracto vocal por uma linha de transmissão ressonante. Este modelo, embora pareça mais adequado no tratamento das ressonâncias do tracto vocal, é muito mais complexo. Assim, é mais difícil retirar conclusões simples na relação entre as configurações do tracto vocal e as frequências obtidas nas ondas acústicas.

---

[1] Os bordos das cordas vocais encontram-se separados

[2] O modelo de produção de fala considera uma estrutura geométrica que pode ser representada por um conjunto de tubos.

No modelo da cavidade faríngea representam-se as cavidades do tracto vocal através da concatenação de uma série de tubos ressonantes de distintas secções, e confronta-se o efeito da variação do volume ou área das aberturas de cada um dos tubos com as ressonâncias globais resultantes. Nestes tubos, o aumento do seu volume leva à diminuição da sua frequência de ressonância, e o aumento da área leva, por seu lado, ao aumento dessa frequência. No entanto, quando se unem dois ou mais tubos, cada um afecta o modo de vibração do ar em todos os outros, elevando a complexidade da interpretação das frequências de ressonância individuais.

1.2.2.1 O Tubo Acústico

Como foi referido anteriormente, todo o som, quando propagado no ar exterior já não é igual ao que foi produzido na fonte, uma vez que já sofreu as alterações impostas pelo tracto vocal. Deste modo, as cavidades superiores do aparelho fonador têm o comportamento de um tubo acústico responsável pela modulação das ondas sonoras provenientes da laringe. Este tubo vai desde a glote[1] até aos lábios. A área da sua secção transversal pode ter até cerca de 20 cm^2 e é determinada pelas posições dos seus elementos articuladores (lábios, língua, maxilar inferior e véu palatino).

Sabe-se que qualquer tubo percorrido por um fluxo de ar, se convenientemente excitado, actua como um ressoador, cuja resposta a uma dada frequência diferente depende fundamentalmente da sua forma e volume interior. Deste modo, como a ressonância de um tubo acústico depende das suas características físicas (volume, forma, etc.), a variação destas leva à alteração do tipo de ressonância imposta pelo tubo ao ar que nele circula. Esta variação do tracto vocal é função do movimento muscular envolvido na actividade dos seus órgãos articulatórios, nomeadamente, a língua, os lábios e o véu palatino.

1.2.2.2 Ressonâncias do Tracto Vocal

A diferença entre o espectro das ondas acústicas propagadas no ar exterior e o espectro das ondas emitidas ao nível da glote, resulta das características de transmissão específicas do tracto vocal. Essas características assemelham-se a um filtro com resposta em frequência complexa, como se pode verificar na Figura 1.3 b). Assim, as componentes das ondas acústicas

[1] Abertura posterior da laringe

pertencentes a determinadas gamas de frequências são transmitidas, enquanto outras, fora dessas gamas de frequências são severamente atenuadas. Deste modo, a resposta em frequência do tracto vocal é caracterizada por uma série de regiões, que se designam habitualmente por formantes, responsáveis pela transmissão das ondas acústicas com a menor atenuação.

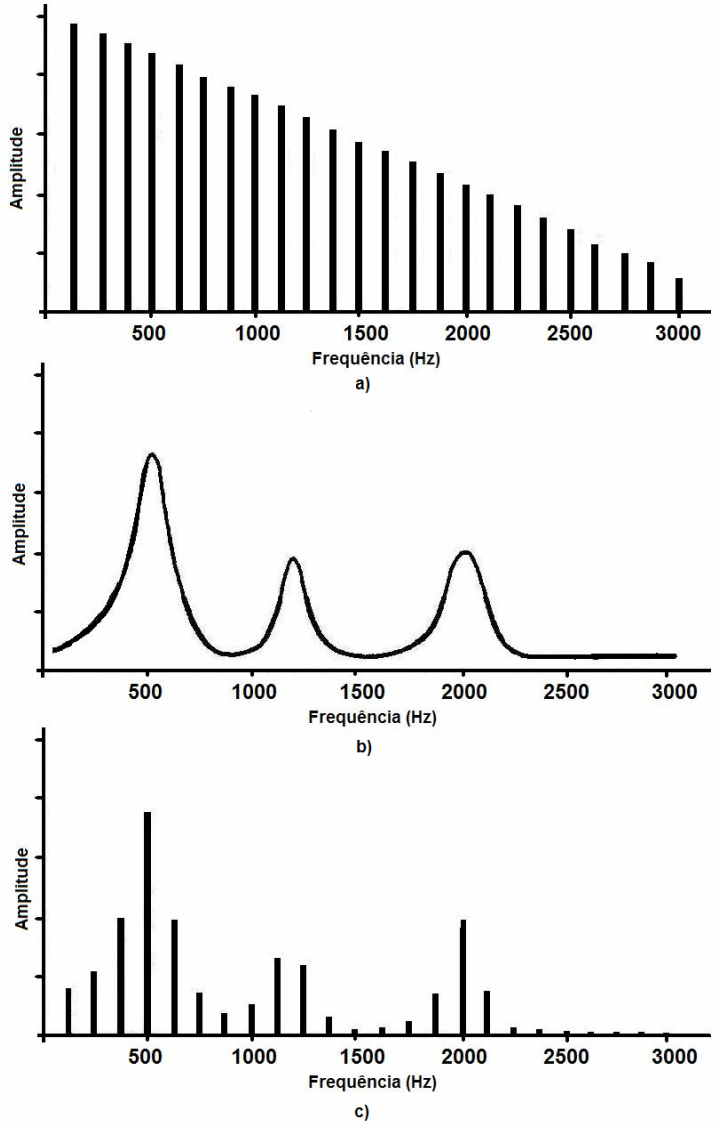


Figura 1.3 - Processo de filtragem por parte do tracto vocal. a) Espectro do sinal de excitação; b) Resposta em frequência do filtro do tracto vocal; c) Espectro do sinal acústico resultante

As[1] formantes estão relacionadas com os picos visualizados no espectro suavizado de um sinal de voz, e são geralmente identificadas pela sua posição ao longo do eixo da frequência, sendo

[1] Também se encontram autores que utilizam o género masculino para denominar estas frequências de ressonância

designadas por, formante 1 (ou F1), formante 2 (ou F2), formante 3 (ou F3), etc. Na Figura 1.3 apresenta-se um exemplo ilustrativo da filtragem efectuada pelo tracto vocal, onde temos em a) o sinal de excitação, em b) a resposta em frequência do tracto vocal, e em c) o espectro do sinal acústico resultante. Normalmente nas configurações do tracto vocal existem entre três e cinco formantes para frequências até 5 KHz, sendo que a localização das três primeiras é abaixo dos 3 KHz, que são as fundamentais para a síntese e percepção da fala. Por seu lado, as formantes de ordem superior são mais importantes na representação da fala não vozeada. No exemplo acima apresentado, são consideradas apenas as três primeiras formantes, F1, F2 e F3.

Por vezes, e embora o efeito ressonante do tracto vocal esteja presente em todos os sons, é difícil a distinção e localização de algumas delas em alguns tipos de sons. Esta situação pode ocorrer quando a fonte de energia acústica de um som particular não forneça energia nas bandas de passagem da função de transferência do tracto vocal, ou quando a potência de excitação nessas bandas for consideravelmente menor que a potência nas outras frequências, de tal modo que as potências das frequências formantes do som produzido não é o suficiente para sobressaírem em relação às restantes.

As frequências formantes permitem identificar certas características físicas dos sons. As zonas de maior concentração de energia estão associadas geralmente a zonas de grande proximidade de duas ou mais formantes.

É importante referir que, sabendo-se que cada configuração distinta do tracto vocal, corresponde, no sinal de voz, a um grupo de frequências de ressonância designadas por formantes, e como essa configuração é variante no domínio temporal, as propriedades espectrais dos sinais de voz também o são, tratando-se então de um sinal não estacionário.

1.2.3 Acoplamento Nasal

Na parte superior da faringe, o fluxo de ar tem a possibilidade de seguir dois caminhos em bifurcação no seu trajecto até ao exterior, os canais bucal e nasal. Existe também o véu palatino, que é fundamentalmente uma membrana dotada de alguma mobilidade, que é capaz de obstruir o acesso ao canal nasal. Assim, quando esta membrana se encontra unida à parede superior da faringe, os sons articulados denominam-se orais, uma vez que é o tracto vocal[1] a única cavidade de ressonância que produz o som. Quando essa membrana se encontra descida, ambas as

[1] Faringe e cavidade bucal

passagens de ar ficam livres, e o tracto vocal é acoplado acusticamente com a cavidade nasal que funciona como segundo canal de passagem e que leva à introdução de ressonâncias adicionais na vibração de ar que a atravessa, produzindo-se assim os sons nasais.

Na análise espectral, a utilização do canal nasal como cavidade de ressonância adicional, leva ao surgimento de anti-ressonâncias e de ressonâncias adicionais, que provocam alterações nas formantes relacionadas com a configuração do tracto vocal, durante a produção dos sons nasalados. Estas alterações consistem na diminuição da energia das formantes[1] e no aumento da largura de banda. O espectro contendo as anti-ressonâncias apresenta potências insignificantes em determinadas frequências, podendo mesmo apresentar potências nulas.

O tracto nasal constitui um caminho auxiliar na transmissão do som. O comprimento típico do tracto nasal num adulto masculino é de 12cm. O acoplamento entre os tractos vocal e nasal é controlado pelo tamanho da abertura do véu palatino. Esta abertura pode variar desde os zero aos $5cm^2$, no caso masculino [3].

1.2.4 Tipos de Excitação

Como já foi claramente referido, a fala é composta por ondas acústicas provenientes do sistema fonador. Os pulmões são os responsáveis pelo fornecimento da potência de excitação no sistema acústico. Podemos dividir a produção das ondas acústicas em três mecanismos diferentes:

- Formar uma constrição ao nível da cavidade bucal, de tal modo que forçando o ar a passar por esse estreitamento com uma dada velocidade, se produz turbulência;
- Obstruir completamente a cavidade bucal de forma a gerar-se uma elevada pressão atrás dessa obstrução que ao ser anulada abruptamente, produza uma excitação transitória;
- Forçar o ar a atravessar a glote com a tensão das cordas vocais ajustada de modo a vibrarem, o que leva à formação de um trem de impulsos quase periódico que levam à excitação do tracto vocal.

Consoante o tipo de excitação, podemos ter sons vozeados ou não vozeados. Esta diferença tem a ver com a vibração, ou ausência dela, ao nível das cordas vocais, respectivamente. Os sons vozeados são sinais aproximadamente periódicos no domínio dos tempos. O período do sinal

[1] Em especial da primeira formante (F1)

corresponde à frequência denominada *pitch*[1], que também se designa por frequência fundamental.

No domínio das frequências, os sons vozeados caracterizam-se por linhas espectrais, ou harmónicas, e apresentam uma estrutura de formantes nítida. Apresentam ainda uma componente de baixa frequência que se manifesta nos espectrogramas sob a forma de uma barra horizontal próxima do eixo horizontal. Por outro lado, os sons não vozeados correspondem a sinais de natureza aleatória, no domínio dos tempos, apresentando um espectro de elevada largura de banda. É de referir que as zonas não vozeadas de um sinal de fala têm em geral menor energia que as zonas vozeadas.

A estrutura harmónica de um sinal de voz é o resultado da quase periodicidade do sinal, que pode ser atribuída à vibração das cordas vocais. No que diz respeito à distribuição espectral, esta surge da interacção entre a fonte de excitação e o tubo acústico, formado pelo tracto vocal e nasal (no caso dos sons nasalados).

1.2.5 Classificação fonética

Como já referido anteriormente, podemos dividir os sons produzidos em três grupos principais:

- Vozeados: se o tracto vocal for excitado por pulsos quase periódicos de pressão de ar causada pela vibração das cordas vocais;
- Fricativos: produzidos por formação de uma constrição em qualquer zona do tracto vocal, criando assim turbulência que produz uma fonte de ruído que vai excitar acusticamente o tracto vocal. Este tipo de sons pode apresentar algumas características semelhantes aos vozeados, sendo designados por vozeados, ou todas distintas, sendo então designados por não vozeados;
- Plosivos: criados pelo fecho total do tracto vocal, provocando assim uma pressão de ar, e subsequente libertação rápida.

[1] Grandeza relativa à frequência percebida. Há autores que utilizam f_0 para representar o valor objectivo dessa frequência de vibração e reservam *pitch* para a sua caracterização perceptual [4]

Os segmentos fonéticos associados a cada som além de se distinguirem pela presença ou ausência de vozeamento, diferem no modo de articulação, levando deste modo a serem divididos em classes diferentes (vogais, glides, oclusivas, fricativas, nasais e líquidas) [5]. Dentro de cada classe esses segmentos fonéticos podem ainda ser distinguidos pelo ponto de articulação no tracto vocal. Para representar os segmentos fonéticos é utilizado um alfabeto próprio, existindo por exemplo o alfabeto fonético IPA[1], ou o alfabeto fonético SAMPA[2]. O SAMPA é presentemente o alfabeto mais utilizado para transcrever *corpus*[3] de fala para o Português Europeu. Na Tabela 1.1 são apresentados os subconjuntos dos alfabetos IPA e SAMPA que representam o Português Europeu comum.

Tabela 1.1 **-** Alfabetos IPA e SAMPA de descrição do Português Europeu e caracterização dos respectivos segmentos fonéticos pela presença de vozeamento, tipo e posição de articulação no tracto vocal [5]

Classe	símbolo IPA	símbolo SAMPA	Altura da elevação da língua	Posição da língua na cavidade bocal
Vogais	ɐ	6	média	média
	a	a	baixa	média
	e	e	média	anterior
	ɛ	E	baixa	anterior
	ɨ	@	alta	média
	i	i	alta	anterior
	o	o	média	posterior
	ɔ	O	baixa	posterior
	u	u	alta	posterior
	ɐ̃	6~	média	média
	ẽ	e~	média	anterior
	ĩ	i~	alta	anterior
	õ	o~	média	posterior
	ũ	u~	alta	posterior
Glides	w	w	alta	posterior
	j	j	alta	anterior
	w̃	w~	alta	posterior
	j̃	j~	alta	anterior

[1] É mais conhecido por alfabeto fonético internacional (IPA – *International Phonetic Alphabet*).

[2] *SAM Phonetic Alphabet*

[3] Base de dados de sinais de fala, utilizado para desenvolvimento de sistemas e aplicações em processamento da fala

Classe	símbolo IPA	símbolo SAMPA	Presença de Vozeamento	Ponto de articulação
Oclusivas	p	p0,p	não	bilabial
	t	t0,t	não	apicodental
	k	k0,k	não	velar
	b	b0,b	sim	bilabial
	d	d0,d	sim	apicodental
	g	g0,g	sim	velar
Fricativas	f	f	não	labiodental
	s	s	não	apicodental
	ʃ	S	não	palatal
	v	v	sim	labiodental
	z	z	sim	apicodental
	ʒ	Z	sim	palatal
Nasais	m	m	sim	bilabial
	n	n	sim	apicodental
	ɲ	J	sim	palatal
		N	sim	
Líquidas	l	l	sim	apicodental
	ɫ	l~	sim	apicodental
	ʎ	L	sim	palatal
	R	R		velar
	ɾ	r		apicodental
Silêncio		sil		

Os sons correspondentes às vogais são normalmente vozeados e produzidos com o tracto vocal numa forma definida, quasi-estática. Para o Português Europeu existem 9 vogais não nasais e 5 vogais nasais. As vogais apresentam geralmente uma duração maior do que as glides e consoantes e uma melhor definição em frequência. Uma característica que se verifica com bastante frequência no Português Europeu é o fenómeno de redução vocálica, que se caracteriza pela diminuição[1] ou mesmo supressão, de um segmento vocálico.

Os sons correspondentes às glides ou semi-vogais, e os respectivos sons nasalados, ocorrem em Português Europeu simultaneamente com uma vogal precedente ou procedente, formando ditongos, em que há transição das formantes entre dois valores, correspondentes aos dois sons distintos do ditongo. As glides podem ser vistas como vogais com maior constrição e menor duração que as vogais respectivas.

[1] Da energia e duração

Os sons correspondentes às oclusivas são sons produzidos pela constrição total do tracto vocal (na zona de oclusão), seguida da libertação da pressão acumulada (na zona de explosão). As diferentes oclusivas são distinguidas através do ponto em que se dá a oclusão e da presença ou ausência de vozeamento.

Os sons correspondentes às fricativas são produzidos com uma constrição do tracto vocal, que dá origem a turbulência. As fricativas podem ser distinguidas através do ponto de constrição e da presença ou ausência de vozeamento. As fricativas vozeadas, apresentam uma componente não periódica, sendo consideradas como tendo excitação mista. Uma das características das fricativas, contrariamente à maioria das outras classes fonéticas, é a grande energia contida nas altas-frequências. Tal como as oclusivas, as fricativas têm uma intensidade bastante mais baixa que as vogais.

Os sons correspondentes às consoantes nasais são produzidos com vibração das cordas vocais e com o tracto vocal totalmente fechado num ponto ao longo da cavidade bucal. Adicionalmente o véu palatino baixa e, consequentemente, o ar proveniente dos pulmões transita através das narinas. A cavidade bucal embora fechada mantem-se acoplada à faringe e à cavidade nasal, resultando uma anti-ressonância, muitas vezes dominante e cuja frequência é inversamente proporcional à dimensão da constrição da cavidade bucal.

Os sons correspondentes às líquidas apresentam espectros com uma estrutura marcada de formantes, embora com uma menor energia. Estas dividem-se em laterais e vibrantes.

É importante referir que os segmentos fonéticos não ocorrem com a mesma frequência.

No Português Europeu, quase todos os sons são produzidos na expiração. Sendo apenas produzidos na inspiração os cliques (ruídos produzidos nas pausas para inspiração) e os silêncios[6].

1.2.6 A Audição da fala

O processo de audição da fala consiste normalmente na recepção das ondas sonoras do ar e sua transformação para que possam ser interpretadas pelo cérebro[1]. O processo de audição

[1] As ondas sonoras podem ser recebidas por outros meios que não o ar, como é o contacto directo com outros elementos de vibração

transforma as ondas sonoras recebidas pelo ouvido em actividade neuronal ao longo do nervo coclear. Existem modelos que tentam explicar a forma como a percepção da fala[1] se processa ao nível do sistema nervoso central apesar de ser um processo complexo [7].

O ouvido humano é o órgão fundamental da audição e pode ser dividido em externo, médio e interno. Cada uma destas regiões desempenha uma função específica no processo de audição. Na Figura 1.4 apresenta-se um esquema do ouvido humano, onde se encontram definidas as referidas regiões, as quais desempenham as seguintes funções:

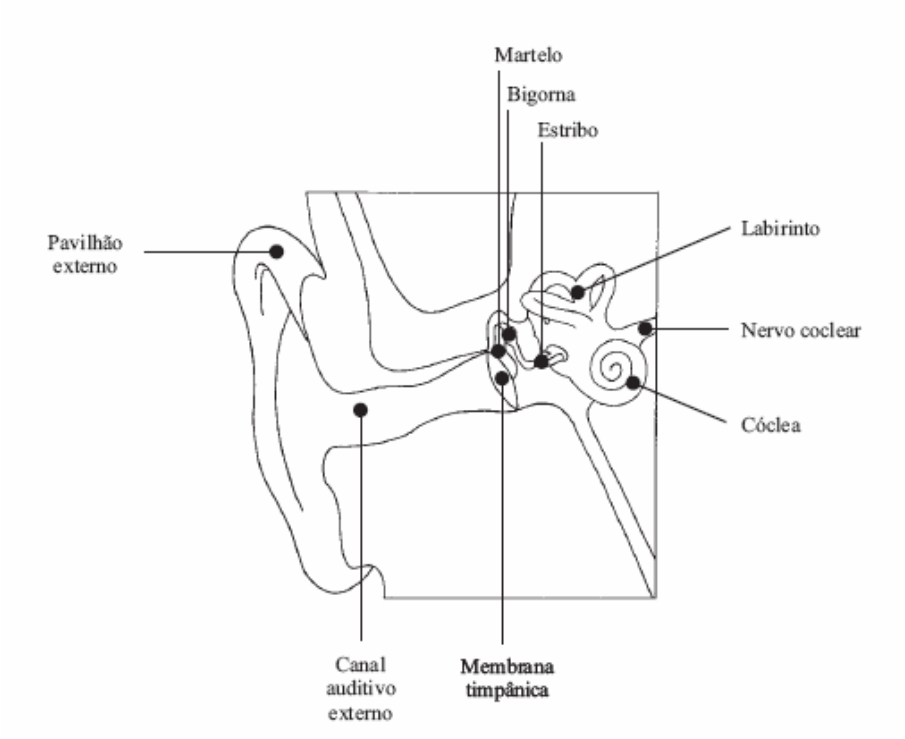


Figura 1.4 - Modelo esquemático do ouvido humano

- Ouvido externo (consiste no pavilhão externo e no canal auditivo externo): capta as ondas sonoras do ar, e, através da sua propagação ao longo do canal auditivo externo, transmite-as ao ouvido médio.
- Ouvido médio (martelo, bigorna e estribo): através da vibração da membrana timpânica, transforma a onda sonora numa vibração mecânica dos ossículos.
- Ouvido interno (composto pela cóclea inserida numa cavidade óssea chamada labirinto e que se encontra ligada ao nervo coclear): as vibrações mecânicas dos ossículos do ouvido

[1] Fase de interpretação e compreensão da mensagem associada às ondas sonoras

médio fazem com que o estribo actue, na entrada da cóclea, sobre a janela oval, que é a membrana que tapa uma extremidade da câmara vestibular, criando uma onda de pressão na substância líquida da cóclea que, por sua vez, faz com que a membrana basilar vibre.

As ondas de pressão são fracas no início da cóclea, tornando-se mais intensas quando atingem a zona da cóclea onde a menbrana basilar tem a frequência natural de ressonância igual à da vibração incidente. É nessa região que a energia da onda é transmitida à membrana e conclui a sua propagação através da cóclea até à janela circular. É conseguida uma dispersão de frequências pela membrana ao longo do seu comprimento, uma vez que as altas frequências estão relacionadas com a base da cóclea, enquanto que as baixas frequências estão relacionadas com a extremidade da membrana. A membrana basilar caracteriza-se por um conjunto de respostas em frequência, do tipo filtro passa-banda, "grosso modo", que varia ao longo da membrana.

Encontra-se distribuído nesta, um conjunto de sensores que tem como função a conversão do movimento mecânico para actividade neuronal. Assentado sobre a superfície da membrana basilar, encontra-se o órgão de Corti. Este órgão converte o movimento mecânico ao longo da membrana basilar sentido pelos sensores em impulsos nervosos em resposta às vibrações da membrana basilar. O órgão de Corti encontra-se ligado ao nervo coclear[1], que conduz ao sistema nervoso central. A partir do nervo coclear, o conhecimento sobre a forma como a informação (actividade neuronal ao longo desse nervo) é processada e interpretada pelo cérebro é bastante reduzida, embora, como foi referido anteriormente, existem já alguns modelos que o tentam explicar.

1.3 A Fala Visual

A face humana é talvez o primeiro meio de comunicação entre humanos. Desde que nascemos, e muito antes de aprendermos a falar, a "comunicação" entre filhos e pais é fortemente influenciada pelas imagens que vamos guardando. Esta capacidade não é perdida. Através da face é facilmente identificado o estado psicológico de uma pessoa, podendo ser uma característica importante para o contexto de uma determinada comunicação. Pode então ser dito,

[1] Ou auditivo

que tal como a fala, a face pode desempenhar uma função importante nas formas de comunicação do ser humano. Através da fala o ser humano exterioriza os seus pensamentos, mas sem nunca esquecer que a face poderá jogar um papel decisivo, daí o ditado "às vezes uma imagem vale mais que mil palavras". Dada a familiaridade da face humana, e a sua função relevante na comunicação desde o momento em que nascemos, tornou-se desafiador a avaliação da sua importância na comunicação oral humana.

Estudos realizados por diversos autores revelam um aumento significativo da compreensão da fala se o som é acompanhado por um padrão visual, cujos movimentos são coordenados por um texto falado.

Como foi referido, o ser humano produz a fala em certos órgãos articuladores (secção 1.2.1) que são visíveis, nomeadamente os lábios, a língua e os dentes. No entanto, a fala contínua não significa som contínuo, uma vez que em intervalos de silêncio, existem alguns gestos que antecipam o som seguinte. Deste modo, na fala, temos partes que apenas são visíveis, partes que apenas são audíveis e partes com ambas características em simultâneo. Este é um dos motivos que levaram a assumir a comunicação falada de forma bimodal, uma vez que a informação é transmitida pelo canal acústico e visual (ou óptico).

Durante a fala, os movimentos visuais devem-se à acção dos músculos da face e maxilar. Os músculos mais importantes são os que provocam o movimento dos lábios pois eles modulam os sons falados. Na Figura 1.5 e Figura 1.6 são apresentados os músculos que provocam os movimentos dos lábios e do maxilar inferior, em todas as direcções, de modo a poderem modular os sons pretendidos.

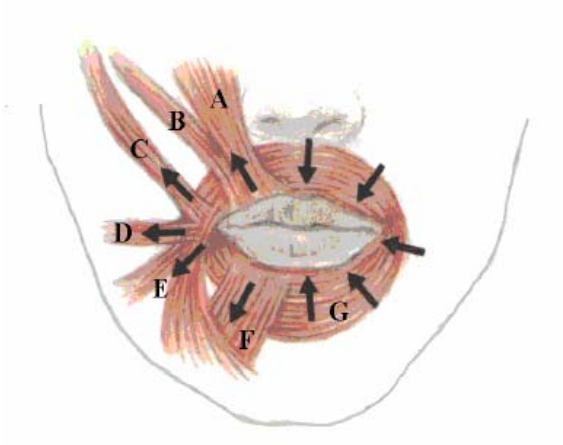


Figura 1.5 - Músculos labiais envolvendo a boca e direccionamento da sua contracção (A – *levator labiisuperioris*, B – *zygomaticus minor*, C - *zygomaticus major*, D - *risorius*, E – *depressor anguli oris*, F - *labii inferioris*, G - *orbicularis oris*)

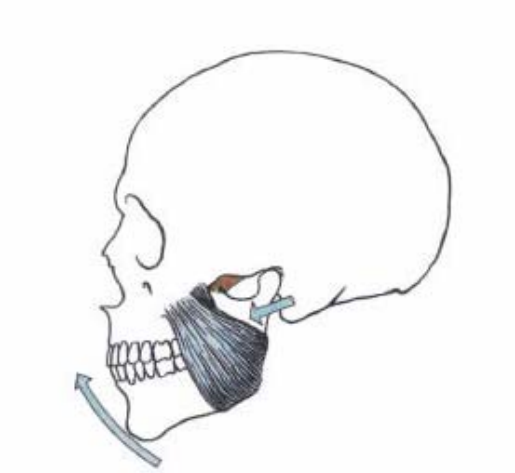


Figura 1.6 - Músculo *masseter* e movimentos do maxilar inferior

Os músculos *levator labii superioris*, *zygomaticus minor*, *zygomaticus major*, *levator angulis ori* e *risorius* são utilizados para o movimento do lábio superior. Os músculos *levator labii superioris* e *musculus zygomaticus minor*, contribuem para a mesma função, principalmente em levantar a parte média do lábio superior. Estes músculos são também os responsáveis por levantar e mover os cantos dos lábios. O músculo *risorius* tem como função apenas mover os cantos dos lábios.

Os músculos *depressor anguli oris* e *depressor labii inferioris* são responsáveis pelo movimento do lábio inferior. O músculo *depressor anguli oris* move o canto da boca para baixo e lateralmente. O músculo *depressor labii inferioris* move a parte media do lábio inferior ainda mais para baixo. O músculo *masseter* é responsável pelo movimento do maxilar inferior para cima e para baixo [8][9].

Na componente visual da comunicação falada, não são todavia apenas os movimentos dos lábios que ajudam à compreensão do que é falado. Zonas como a dos olhos, em especial a zona por cima dos olhos[1] e a zona do queixo contêm informação discriminante em função dos sons produzidos. Na Figura 1.7 é apresentada a divisão da cabeça humana em regiões. Esta divisão permite dividir em zonas com maior ou menor actividade durante a fala, ou seja, permitem orientar a análise para regiões de interesse onde as variações são mais notórias.

As regiões da cabeça que geralmente apresentam maiores variações durante a comunicação através da fala são as da face, nomeadamente a da boca (*régio oralis*), mas também a zona dos olhos (*régio orbitalis*), testa (*régio frontalis*) e queixo (*régio mentalis*).

[1] Sobrancelhas ou sobrolhos

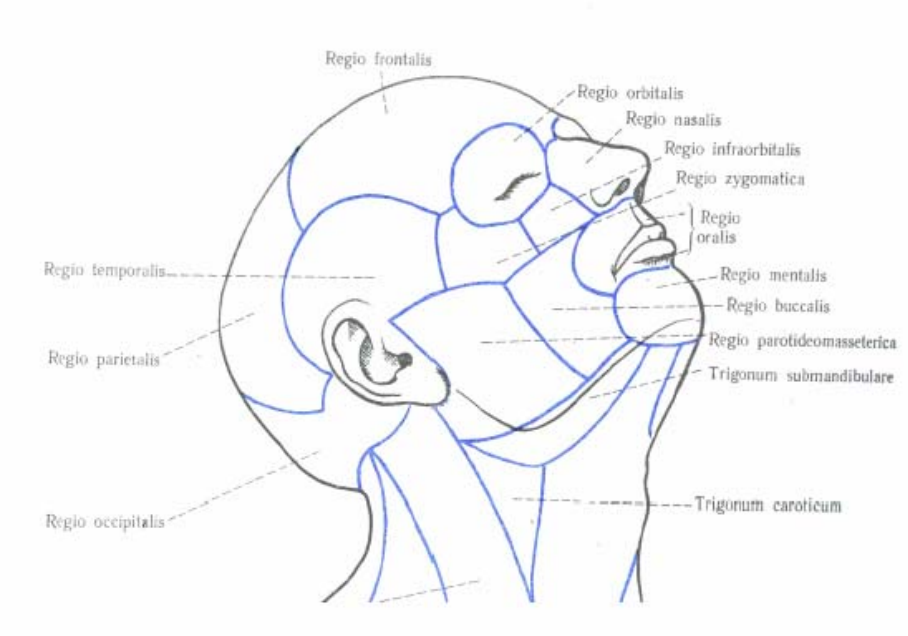


Figura 1.7 - Áreas da cabeça humana

1.3.1 Visemas

No sinal de fala acústico, o fonema é a unidade linguística de som mais pequena que se consegue identificar, usando a correspondência letra-som. De modo análogo, pode ser descrito um visema[1] como a unidade mais pequena em que se pode dividir a fala, a nível de características visuais.

O termo visema foi pela primeira vez utilizado por C. G. Fisher, no contexto do processamento da fala, para descrever "alguma individual e contrastada unidade visual reconhecida" [10]**Error! Reference source not found.**.

No Português, tal como em outras linguagens, existem muitos sons que são visualmente ambíguos. Por exemplo os fonemas /p/ e /b/ são articulados do mesmo modo, com a boca fechada, o que os torna visualmente iguais. Este tipo de fonemas é agrupado na mesma classe de visemas. O termo visema não é o único termo utilizado para descrever fala visual. Os

[1] *Visual phoneme* (fonema visual)

investigadores Jeffers e Barley utilizaram o termo *speechreading movement*[1] para descrever qualquer padrão visual usualmente comum a dois ou mais sons [11]. Ambas designações, visema e *speechreading movement*, são sinónimas e usadas sem diferenciação na mais variada literatura da área. No caso deste trabalho, uma vez que a unidade de fala utilizada é a palavra, é importante referir a ambiguidade visual a este nível. Palavras como "cinco" e "oito", por exemplo, não são visualmente diferenciáveis. Os sons que apresentam iguais movimentos dos lábios de tal modo que não conseguem ser distinguidas apenas pelas características visuais podem ser agrupadas em classes diferentes. Os elementos da mesma classe[2] correspondem aos mesmos visemas, mas um visema não é necessariamente idêntico a uma dessas classes de palavras.

Por outro lado, a importância do contexto linguístico foi estudado por A. G. Bell. Este investigador constatou que muitos sons, sendo invisíveis quando falados isoladamente e apresentando o mesmo movimento dos lábios, se tornam diferenciáveis visualmente quando inseridos num contexto. A sua pesquisa mostrou que as palavras pequenas, como por exemplo as conjunções e pronomes, são as mais difíceis para a leitura labial. Estes resultados formam confirmados posteriormente, em 1980, por Pickett [12].

A fala não é vista apenas como um conjunto de fonemas distintos agrupados, mas também por um conjunto de visemas associados. Diversos investigadores, tal como Liberman, Cooper, Shankweiler, Studdert, demostraram que um fonema é significativamente influenciado pelos seus vizinhos e pela sua localização temporal no segmento de fala [13]. Deste modo, a análise da concatenação dos visemas, especialmente para sistemas de síntese de fala visual, tornou-se um ponto de investigação importante. O caminho seguido foi a compreensão do efeito de co-articulação. Benguerel e Pichora-Fuller [14] definiram a co-articulação como a alteração de um conjunto de movimentos articulatórios realizados na produção de um fonema por aqueles realizados na produção de um adjacente ou de um fonema vizinho. Montgomery foi um dos primeiros investigadores a propor um modelo para a co-articulação [15].

Os efeitos da co-articulação têm sido o principal obstáculo ao estabelecimento de um conjunto de grupos de visemas universal. É importante ter em conta os seus efeitos em sistemas de síntese de fala visual (sistemas de animação facial), caso contrário a inteligibilidade da fala pode ficar comprometida.

---

[1] Leitura labial

[2] Em [16] o autor designa essas classes por homophonous, no entanto, devido à nomenclatura da palavra utilizada, essa terminologia não foi adoptada por não parecer a mais correcta.

Neste trabalho, uma vez que a unidade básica de fala utilizada no desenvolvimento do sistema foi a palavra e não o fonema, não foi realizado um estudo do conjunto de visemas presentes nas palavras do vocabulário, para o Português Europeu. Um estudo dos visemas poderia ser importante na fase de desenvolvimento do sistema de reconhecimento de fala baseado nas características visuais, uma vez que permitiria identificar os erros ocorridos nas palavras e analisá-los segundo os visemas que as constituem.

A título de exemplo é apresentado na Tabela 1.2, o agrupamento dos fonemas num conjunto de visemas para a língua inglesa. Actualmente, e apesar do esforço de vários investigadores, não existe nenhum sistema de visemas universal que englobe todos os fonemas presentes numa comunicação visual [17]. Em [17] são apresentados quatro estudos publicados, de conjuntos de visemas para a mesma classe, para a língua Inglesa. Pode ser constatado que alguns dos visemas existem nas várias tabelas (/p, b, m/ e /f, v/) e podem ser consideram universais, para a língua inglesa, enquanto que os outros variam, dependendo especialmente do orador e suas condicionantes linguísticas.

TABELA 1.2 - AGRUPAMENTO DOS FONEMAS EM VISEMAS PARA A LÍNGUA INGLESA USANDO A NOTAÇÃO SAMPA

Visema	Fonema
0	sil
1	f v w
2	s z
3	S Z
4	p b m
5	g k x n N r j
6	t d
7	l
8	I e:
9	E E:
10	A
11	@
12	i
13	O Y y u 2: o: 9 9: O:
14	a:

1.4 O reconhecimento automático da fala

O reconhecimento automático de fala consiste na transcrição dos elementos recebidos, sinal acústico e visual, numa palavra ou sequência de palavras. Os sistemas de reconhecimento de fala desempenham o papel do ouvinte, convertendo o sinal acústico e visual recebido numa mensagem escrita ou identificando uma instrução de comando [18].

Embora tenha começado por consistir numa tarefa de classificação de um conjunto de amostras do sinal de fala, o reconhecimento da fala, hoje em dia, tornou-se num processo complexo, onde são incluídos um conjunto diversificado de conhecimentos, de modo a classificar correctamente fala espontânea para grandes vocabulários. Os conceitos associados ao processo de reconhecimento serão apresentados no capítulo 2.

De seguida vão ser apresentadas as principais dificuldades associadas aos sistemas de reconhecimento de fala tradicionais, sendo apontados alguns dos métodos existentes para as solucionar.

1.4.1 Dificuldades

Sendo a fala um processo de comunicação, natural no Homem, uma vez que é o resultado de um processo de aprendizagem, treino e aperfeiçoamento, desde o nascimento, podia ser pensado que a implementação de um sistema de reconhecimento de fala seria uma tarefa simples. Apesar de todos os esforços no desenvolvimento e implementação das mais variadas estratégias, o processo de reconhecimento automático da fala sempre se comportou e ainda hoje em dia se caracteriza por ser um processo complexo.

A grande dificuldade deve-se principalmente a dois motivos, a grande variabilidade associada a cada unidade de fala, assim como a dimensão temporal associada à sucessão das unidades de fala.

É importante referir que um sinal de fala é muito dependente do orador, mas, para o mesmo orador, também se verifica uma variabilidade dependente das suas condições, quer físicas quer psíquicas, entre outras. Por exemplo, a intensidade e a velocidade, são factores que podem apresentar uma grande variabilidade.

Uma vez que os sinais de fala são geralmente contínuos, a localização das unidades de fala (determinação das fronteiras temporais) nesse sinal acústico, através de meios automáticos é decisivo no crescimento da complexidade dos sistemas RAF. Se se considerar somente os sinais contínuos, problemas relacionados com a co-articulação dos sons podem levar à assimilação e principalmente, à supressão de alguns sons.

É importante não esquecer que associado a um sistema de reconhecimento, existe um vocabulário a ser interpretado. O tamanho deste vocabulário influencia decisivamente na complexidade do reconhecedor e pode ter um peso significativo no seu desempenho. No entanto, só por si, o tamanho do vocabulário não implica directamente a complexidade da tarefa, sendo mais importante para avaliar a tarefa as características das palavras desse vocabulário.

Existem vários tipos de ruído que podem interferir em diferentes níveis do processo de produção/reconhecimento da fala. O meio de comunicação atravessado pelo sinal de fala não é imune a diversos problemas que podem surgir. Desde o orador ao sistema de reconhecimento, tem que se prestar grande atenção ao elemento de recolha do sinal de fala (microfone, telefone, etc.), e ao ambiente que envolve o orador. Todos estes elementos podem introduzir no sinal de fala uma forma de ruído ou distorção (ver Figura 1.8). Em ambientes de desenvolvimento (laboratórios), estes factores tendem a ser minimizados através do uso de microfones de alta qualidade e a ambientes limpos[1]. No entanto tem sido objectivo primordial tentar desenvolver aplicações de reconhecimento de fala em ambientes o mais próximos da sua utilização real.

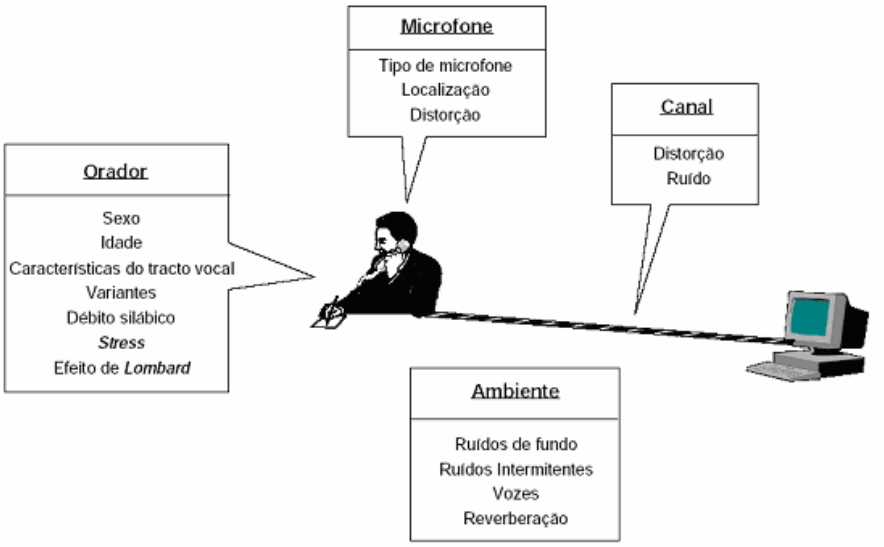


Figura 1.8 - Factores que podem afectar o desempenho de um sistema de reconhecimento de fala [19]

[1] Ambientes em condições de ruído controlados

Da análise da Figura 1.8 podemos observar a diversidade de condicionantes associado a cada um dos elementos que podem interferir no desempenho de um RAF. O ambiente é sem dúvida um dos principais factores a considerar e o que pode mais seriamente permitir um baixo desempenho, devido à grande variabilidade das perturbações que poderão estar em causa. Existem os ruídos de fundo, que apresentam uma forma contínua e estrutura diversa. Existem os ruídos intermitentes como o bater de uma porta ou ainda ruídos produzidos pelo orador (tosse, ruídos bucais, etc.) ou outros oradores (vozes). Além dos ruídos referidos, o ambiente também pode ser responsável por distorções devidas às propriedades acústicas da sala. A reverberação é um dos exemplos mais comuns deste tipo de distorção.

Outros dos factores importantes é a diferença entre oradores e estilos de produção de fala (sistemas multi-orador). Neste sentido há que ter em atenção factores como o sexo, a idade, variantes dialectais, entre outras, sem esquecer que o estilo da fala de um orador pode ser dependente das condições, físicas e psicológicas, ou do próprio ambiente, em que este se encontra, que pode levar a alterações de comportamento como em casos de stress ou o chamado efeito de Lombard. O efeito de Lombard caracteriza-se pela alteração do mecanismo de produção de fala do orador [20], que leva à alteração da localização dos formantes, aumento da duração das vogais, entre outras. Este é um dos factores mais difíceis de modelar, uma vez que apesentam uma grande variabilidade de orador para orador.

O tipo e localização do elemento de recolha do sinal de fala (geralmente o microfone) podem degradar a qualidade do sinal, que a nível de características como a amplitude, quer a nível da distorção inserida devido à resposta em frequência do próprio dispositivo.

Como foi referido, um factor que pode ser decisivo na taxa de acerto dos reconhecedores automáticos de fala é o ambiente acústico em que estes operam. Um ambiente acústico adverso pode constituir um grande obstáculo para o melhor dos reconhecedores. Para muitos investigadores da área, actualmente este é o grande obstáculo e maior desafio. É neste sentido que caminha esta dissertação. A utilização das características visuais no processo de reconhecimento automático da fala, visa robustecer o reconhecimento em ambientes acústicos, uma vez que estas características, embora de uma forma variável, complementam as características acústicas.

Não pode deixar de se referir que a abordagem seguida nesta dissertação é apenas mais uma abordagem ao robustecimento do sistema de reconhecimento automático de fala. Esta técnica é

relativamente recente e daí um dos interesses na sua implementação. No entanto existem outras técnicas, desenvovidas e com resultados comprovados, com o objectivo de conferir mais robustez no reconhecimento automático de fala, baseados fundamentalmente nas características do sinal acústico. De seguida é apresentada uma breve referência a essas técnicas.

1.4.2 Abordagens ao Reconhecimento Robusto da Fala

Um dos principais problemas do bom desempenho dos sistemas de reconhecimento automático de fala é a presença de ruído[1] acústico no seu ambiente de funcionamento real. Este problema surge normalmente associado às diferenças registadas entre a fase de treino e a fase de teste do sistema. Um sistema é considerado robusto se conseguir manter um bom desempenho mesmo quando as diferenças entre as condições de treino e teste forem diferentes.

Alguns investigadores acreditam que esse problema pode ser resolvido aumentando o conjunto de amostras de treino de modo a incluir todas as variações possíveis [21]. Deste modo os modelos gerados deveriam ser capazes, através do aumento da sua complexidade de acomodar essa variabilidade, mantendo ou aumentando a capacidade discriminativa. Embora possa melhorar o desempenho de alguns sistemas, essa consideração não é completamente verdadeira, uma vez que os modelos calculados com um conjunto tão diversificado e enorme de dados poderiam tornar-se difusos e diluídos, ou seja, iriam apresentar uma elevada variância, o que os tornaria pouco precisos na classificação correcta dos sinais de fala.

Apesar dos inúmeros esforços para melhorar a robustez dos reconhecedores de fala actuais quanto ao ruído, o problema persiste uma vez que muitos desses reconhecedores assumem, no seu desenvolvimento, níveis de ruído baixo ou então modelam o ruído como se fosse do tipo aditivo estacionário branco-gaussiano[2] (AWGN) ou do tipo ruído cor-de-rosa[3]. Esta situação pode originar uma degradação muito grande do desempenho desses sistemas quando testados no mundo real. O principal obstáculo para o desenvolvimento dos sistemas de reconhecimento em ambientes de funcionamento reais prende-se com as bases de dados. Desenvolver uma base de dados para treino e teste em condições de funcionamento final do sistema torna-se extremamente dispendioso [22].

---

[1] Ruído aditivo

[2] Ruído branco é um ruído cuja potência está igualmente distribuída em todas as frequências, considerando uma escala natural.

[3] Ruído cor-de-rosa é um ruído cuja potência está igualmente distribuída em todas as frequências, considerando uma escala logarítmica.

Para resolver o problema da robustez, têm que ser entendidas as características básicas do sinal de fala, assim como o efeito da interferência das diferentes fontes de distorção ou variabilidade que podem afectar esse sinal. Adquirido esse conhecimento, é possível implementar técnicas que visam todo o sistema de reconhecimento, podendo ser implementadas na aquisição do sinal ou na fase de modelação acústica [23][24][25][26].

No que diz respeito à aquisição de sinal de fala, existem algumas técnicas relacionadas com o número, posição e qualidade de microfones e com algoritmos de cancelamento de elementos adversos. No entanto, os métodos mais utilizados visam a extracção de parâmetros acústicos robustos que se seguem à obtenção do sinal. De igual importância existem técnicas que permitem efectuar uma normalização do sinal de fala assim como técnicas que operam ao nível dos modelos acústicos.

As inúmeras técnicas existentes na abordagem ao reconhecimento robusto da fala têm sido enquadradas em diferentes grupos, segundo a metodologia seguida. A divisão mais utilizada enquadra-as em grupos diferentes como são a extracção de características resistentes ao ruído acústico, normalização de parâmetros acústicos e técnicas de normalização e adaptação dos modelos[1]. Ultimamente têm surgido novas abordagens que constituem variantes às abordagens que vão ser apresentadas, assim como novas combinações dessas técnicas especialmente das técnicas de compensação (*Supervised-predictive noise compensation* [27], entre outras). Deve ainda ser feita referência às técnicas multi-canal que foram bastante exploradas na década de 1990.

De seguida vai ser realizada uma breve referência a algumas das técnicas mais utilizadas, referindo desde já que existem muitas mais abordagens a este assunto, mas que apresentam metodologias bastantes semelhantes, uma vez que muitas delas são apenas variações das apresentadas (Jacobian Adaptation [27], entre outras).

1.4.2.1 Selecção e extracção de características resistentes ao ruído acústico

A selecção das características acústicas adequadas é um factor decisivo no desempenho de um sistema de reconhecimento de fala. A sua escolha deve ser feita tendo em mente os seguintes critérios:

[1] Conseguido geralmente através da combinação de modelos paralelos

- Devem conter a máxima informação necessária para o reconhecimento;
- Devem ser o mais possível imunes às características do orador, modo da fala, ruído ambiente, distorção do canal, etc.
- Devem permitir ser estimadas com precisão e confiança;
- Devem permitir ser estimadas através de um processo computacional eficiente;
- Devem ter uma consistência física real (devem estar relacionadas com o sistema auditivo humano).

Os algoritmos pertencentes a esta técnica visam obter uma representação do sinal de fala de uma forma que permita uma descodificação mais eficiente. Estas técnicas são especialmente eficientes quando o ruído ambiente considerado é aditivo e a distorção é linear (ver Figura 1.9).

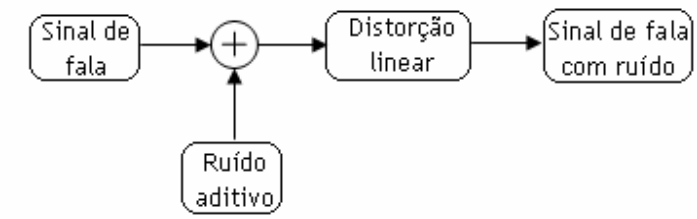


Figura 1.9 - Modelo de um ambiente que introduz ruído aditivo e distorção linear

De seguida são apresentadas algumas destas técnicas.

1.4.2.1.1 *Predição Linear Perceptual*

A técnica de predição linear perceptual (PLP - *Perceptual Linear Prediction*) [28], pretende simular algumas características do sistema auditivo humano. Essa simulação é conseguida através de aproximações práticas sendo o espectro resultante aproximado por um modelo autoregressivo só com pólos.

Com esta técnica pretende-se filtrar o sinal de fala, através de um conjunto de operações que realçam as características mais importantes, dando menos importância às características menos relevantes, o que segundo os autores, não é possível usando predição linear pura e simples, uma vez que esta apresenta resolução espectral idêntica em todas as bandas de frequência. As operações implementadas são:

- Resolução espectral de banda crítica. Realiza-se o ajustamento do espectro de potência do sinal ao longo do eixo da frequência, seguindo-se a convolução com uma curva que simula a banda-crítica (logarítmica a partir aproximadamente dos 1000HZ e uniforme antes). Esta resolução realiza-se numa escala de *Bark*.
- Simulação da diferença registada na sensibilidade da audição humana em diferentes frequências. Para tal é realizada uma pré-enfase do sinal por uma curva de *equal loudness*, traduzida [29] como curva de igual intensidade sonora subjectiva.
- Simulação da relação não linear entre a intensidade do som e a sua percepção subjectiva por compressão da amplitude por uma raiz cúbica.

1.4.2.1.2 *Mel Frequency Cepstral Coefficients* (MFCC)

Existem técnicas para extrair características acústicas baseadas nas características do sistema auditivo humano, como por exemplo, os coeficientes cepstrais definidos na escala de *Mel*, mais conhecidos por coeficientes mel-cepstrais (MFCC). A escolha destes coeficientes tem sido a abordagem preferencial no desenvolvimento de sistemas de reconhecimento da fala. Nesta abordagem é utilizado um banco de filtros passa-banda triangulares espaçados de acordo com a escala de *Mel*. Esta escala, tal como a escala de *Bark*, foi desenvolvida de forma a aproximar a resolução do sistema auditivo humano, ou seja, linear até aos 1000Hz e logarítmica a partir dessa frequência. Os coeficientes cepstrais são obtidos aplicando uma DCT (*Discrete Cosine Transform*) ao logaritmo da energia à saída do banco de filtros. Desta última transformação resultam, por um lado, uma representação mais compacta (porque a DCT comprime a informação espectral nos coeficientes de ordem mais baixa) e por outro lado a descorrelação do sinal (o que permite usar matrizes de covariância diagonais com consequente menor degradação da performance) [30].

Esta técnica vai ser mais aprofundada na secção 2.5.1 uma vez que foi a abordagem seguida no sistema que se vai desenvolver para extracção de características acústicas.

1.4.2.2 Técnicas de Normalização de Parâmetros Acústicos

Este tipo de técnicas é utilizado quando não houve preocupação na selecção e extracção das características acústicas seleccionadas.

1.4.2.2.1 Subtracção espectral

A técnica de Subtracção espectral consiste em subtrair à amplitude do espectro do sinal original, a amplitude do espectro do sinal com ruído, estimada a partir dos segmentos do sinal que não contêm fala [31]. Com esta técnica é conseguida uma redução do ruído estacionário ou quase-estacionário do sinal de fala.

Este método assume que o ruído de fundo (ambiente) se mantém localmente estacionário de tal modo que a sua amplitude, estimada num período de tempo exclusivamente com ruído, é igual ao que poderia ser estimado durante a actividade de fala imediatamente a seguir. Outra consideração desta abordagem é que se houver uma mudança para um novo estado estacionário deverá existir tempo suficiente para uma nova reestimação antes de começar um novo segmento com fala, o que leva à necessidade de incluir um detector de início de fala relativamente eficiente. Finalmente, a qualidade da supressão é aceitável se for removido o efeito do ruído apenas na amplitude espectral e não na fase.

Antes de se proceder à reconstrução do sinal, e após realizada a subtracção, deve ser realizada uma rectificação do sinal de fala de forma a obterem-se sempre amplitudes espectrais positivas.

1.4.2.2.2 Subtracção da média cepstral (*Cepstral Mean Normalization* - CMN)

Este é um dos algoritmos de implementação mais simples. É calculado o valor da média dos vectores de características e de seguida realiza-se a subtracção da média de cada um dos vectores [32][33][155]. Deste modo é reduzida a variabilidade dos dados, e por outro lado é realizada, de uma forma eficiente, a normalização do canal e das características do falante. Este procedimento deve ser aplicado durante o treino e durante o desenvolvimento e teste.

O cálculo da média pode ser de curta duração ou de longa duração, embora se tenham verificado melhores resultados na segunda abordagem [22][33].

O principal problema desta técnica tem a ver com o facto de introduzir um atraso considerável, uma vez que tem que se calcular a média.

1.4.2.2.3 *RelAtive SpecTrA* - RASTA

A técnica RASTA surgiu com o objectivo de tratar os componentes não linguísticos, fixos ou que variam lentamente, do sinal de fala [34].

A técnica é uma extensão à abordagem PLP. São adicionados 3 novos passos após o cálculo da resolução espectral de banda crítica. O objectivo consiste na supressão dos factores constantes em componentes do espectro de curta duração antes da estimação do modelo só com pólos. Inicialmente é realizada uma transformação da amplitude espectral, seguida de uma filtragem da trajectória temporal de cada componente espectral transformado, terminando com uma nova transformação da representação filtrada do sinal.

1.4.2.2.4 *Signal Bias Removal* - SBR

Esta técnica consiste na estimação de um valor[1] a partir do método da máxima verosimilhança [35]. Esse valor é, de seguida removido, de modo a minimizar alguns efeitos indesejados, nomeadamente os introduzidos pelo modelo do ambiente. Os valores que se pretendem estimar e remover devem-se às características do ruído e da distorção que afectam o sinal de fala, e que se podem assumir relativamente constantes.

Este algoritmo está usualmente referênciado para HMM's discretos (embora possa ser modificado para outras estruturas de HMM's) e aplica-se no contexto da quantificação vectorial.

1.4.2.3 Técnicas de Normalização de Modelos

Neste tipo de técnicas, os modelos do ruído e do canal são directamente incorporados no processo de reconhecimento.

1.4.2.3.*1 Parallel Model Combination* - PMC

Esta técnica estabelece a combinação de modelos de fala e de ruído [36][37][38]. Os modelos usados são HMMs *standard* com distribuições de probabilidade Gaussianas. O objectivo desta técnica consiste na estimação dos parâmetros de um conjunto de modelos e fala corrompida por

[1] Bias, Viés

ruído, dado um conjunto de modelos de fala sem ruído (nas condições do canal de teste) e um modelo do ruído de fundo. Para implementar esta técnica é necessária a independência entre a fala e o ruído, que a fala e o ruído sejam aditivos no tempo, que uma simples gaussiana ou uma mistura de gaussianas contenha informação suficiente de modo a ser capaz de representar a distribuição dos vectores de observação, e finalmente, que o alinhamento sector/estado, usado para gerar os modelos da fala sem ruído, não seja alterado pela adição de ruído.

Este método requer um grande esforço computacional sendo preterido para o desenvolvimento de aplicações em tempo real. Outro problema prende-se com a compensação desta técnica ser realizada nos coeficientes estáticos, não sendo possível uma simples combinação quando são utilizados coeficientes dinâmicos [39].

1.4.2.3.2 Compensação

A técnica de compensação mais conhecida é a Compatibilidade Estocástica (*Stochastic Matching*) e consiste na diminuição da incompatibilidade entre os ambientes de treino e de teste [40][41][155]. Considerando, que num conjunto de treino, se tem um vector de características X obtido a partir de um sinal S, e modelos de treino λ_x, e que a partir de um sinal de teste T, obtivermos características Y, com modelos de teste λ_y, está técnica pode ser implementada de dois modos distintos. A primeira abordagem consiste em mapear as características distorcidas Y numa estimativa das características originais X de modo a que se possam usar os modelos de treino λ_x. A segunda abordagem consiste em mapear os modelos originais λ_x nos modelos transformados λ_y que melhor se adequam ao vector de características observado Y.

1.4.2.4 Técnicas de realce multi-canal

As técnicas de realce multi-canal apresentam a vantagem de utilizarem vários sinais de entrada, podendo deste modo realizar estimativas das características do ruído presente [42]. Desse modo, é possível implementar técnicas que realizem o cancelamento de certas componentes do ruído presente [43]. De entre várias abordagens a mais utilizada é o cancelamento adaptativo do ruído.

A técnica Cancelamento Adaptativo do Ruído baseia-se na utilização de um canal auxiliar, conhecido por canal referência, que tem o único objectivo de recolher uma amostra do ruído ambiente. Essa amostra é filtrada através de um algoritmo adaptativo sendo posteriormente subtraído esse resultado ao canal principal, onde se encontra o sinal de fala contaminado com ruído [44].

1.4.3 Aplicações

A principal questão relacionada com a aplicação do reconhecimento da fala prende-se com os benefícios que podem ser retirados da interacção homem-máquina. Sem dúvida que um dos principais benefícios centra-se na possibilidade de controlar o ambiente que nos rodeia. Embora tenham sido diversas as orientações seguidas, têm sido, as empresas, as mais preocupadas em produzir aplicações funcionais motivadas principalmente pelos benefícios económicos que deles podem retirar. Este tipo de sistemas provoca uma diferenciação entre produtos, transmitindo uma imagem forte da empresa uma vez que apresenta uma evolução tecnológica, que terá impacto no mercado (quer para a empresa, quer para o próprio produto).

As aplicações da tecnologia do reconhecimento de fala são bastante diversificadas. Na área das telecomunicações, são de destacar nos sistemas de informação, de acesso a dados ou serviços por via telefónica. Existem também múltiplas aplicações em escritórios, desde sistemas de ditado para edição de documentos até controlo de sistemas informáticos através de comandos por voz. Em ambientes fabris já existem também inúmeras aplicações que utilizam reconhecimento de fala. Em actividades profissionais específicas, como a medicina, o reconhecimento de fala tem sido utilizado para preenchimento de relatórios que utilizam vocabulários específicos.

Com a evolução exponencial a nível da electrónica (mais concretamente da microeléctrónica), a integração de sistemas de reconhecimento de fala tem crescido uma vez que permitiu uma evolução e aperfeiçoamanto desses sistemas. Esta evolução tem sofrido uma expansão para para o domínio audio-visual, tendo sido direccionado grande parte do esforço no desenvolvimento de aplicações para reconhecimento de fala em automóveis.

1.5 Sistemas de Reconhecimento Automático de Fala

Os sistemas de reconhecimento automático de fala tem como função determinar a sucessão correcta de símbolos associados à mensagem presente num sinal de fala[1]. Para realizar este processo é necessário que seja implementado um conjunto de funções, que vão desde a captação do sinal até à sua classificação.

Os blocos principais do sistema são o bloco de pré-processamento, ou análise, e o bloco de reconhecimento, ou classificação. No bloco de pré-processamento, o sinal de fala, após ser recolhido pelos elementos de aquisição de sinal do sistema, é convertido numa representação que permita uma classificação mais adequada por parte do classificador. No segundo bloco é atribuído um conjunto de símbolos às características do sinal de fala vindas do bloco de pré-processamento. Nos sistemas a operar com interface tem que ser considerado um bloco para comunicação entre o classificador e a interface da aplicação.

Vai ser apresentada uma primeira abordagem à estrutura de um sistema de reconhecimento automático de fala, introduzindo desde já o conceito associado ao trabalho desenvolvido, ou seja, que o sinal de fala é um sinal bimodal composto pela combinação do sinal acústico e do sinal visual.

Seja $x_a(t)$ o sinal acústico e $x_v(t)$ o sinal visual, componentes do sinal de fala à entrada do sistema multi-modal.

Assumindo as componentes do sinal de fala definidas num dado intervalo de tempo, o módulo de pré-processamento transforma essas componentes em T vectores de características, cada um de dimensão D_a e D_v, para o sinal acústico e e visual, respectivamente,

$$X_{a1}^{\ T} = \{x_{a1}, x_{a2}, x_{a3}, \ldots, x_{aT}\}, \qquad x_{ai} \in \Re^{Da} \tag{1.1}$$

$$X_{v1}^{\ T} = \{x_{v1}, x_{v2}, x_{v3}, \ldots, x_{vT}\}, \qquad x_{vi} \in \Re^{Dv} \tag{1.2}$$

Este módulo deve extrair, do sinal de entrada, apenas a informação relevante para o processo de classificação. Assim tem-se dois vectores de características com as seguintes componentes,

[1] Se for um reconhecedor de palavras isoladas essa sucessão consiste num só símbolo.

$$x_{ai} = \left(x_{ai}^{1}, x_{ai}^{2}, x_{ai}^{3}, \ldots, x_{ai}^{Da} \right) \tag{1.3}$$

$$x_{vi} = \left(x_{vi}^{1}, x_{vi}^{2}, x_{vi}^{3}, \ldots, x_{vi}^{Dv} \right) \tag{1.4}$$

No módulo de classificação os sucessivos vectores de características são transformados numa sucessão de símbolos (geralmente símbolos linguísticos que não são mais do que representações ortográficas de palavras) pertencentes a um vocabulário Γ que estão relacionados com as classes padrão,

$$W_1^k = \{w_1, w_2, w_3, \ldots, w_k\}, \qquad w_i \in \Gamma_w \tag{1.5}$$

O principal objectivo do classificador é, através de um conjunto de funções, discriminar entre os diversos modelos que a cada momento podem ter dado lugar aos vectores de características presentes. Essas funções estão estabelecidas por um conjunto de regras numéricas, sintácticas, etc., que geram o resultado do classificador.

1.5.1 O Paradigma do Reconhecimento

Os sistemas de reconhecimento automático da fala baseiam-se geralmente em estruturas integradas e na modelação estatística. Um sistema integrado de reconhecimento automático de fala contínua realiza o reconhecimento de padrões integrando numa única saída as várias classificações que podem ocorrer durante o processo. Essas são relativas à classificação que pode ser do sinal acústico, da abordagem gramatical, etc. Assumindo um sistema estatístico, é possível estabelecer um conjunto de regras e simplificações matemáticas que levam à uma classificação óptima rigorosa.

Sendo X_a e X_v duas sucessões de vectores de características extraída do sinal de fala, (1.1) e (1.2), o objectivo é determinar a classificação óptima, ou seja, determinar a frase correspondente a uma sucessão de palavras W_c (1.5) resultante da classificação, que apresente a menor probabilidade de errar.

O objectivo do classificador deve atribuir aos vectores X_a e X_v a classe W_c que apresentar maior probabilidade *a posteriori*. O classificador que estabelece esta regra é conhecido por Bayesiano.

$$P(W_c|X_a, X_v) > P(W_i|X_a, X_v), \quad \forall i = 1,...,C \quad e \quad i \neq c \tag{1.6}$$

Uma vez que pode tornar impraticável, em aplicações mais complexas, estimar as probabilidades *à posteriori*, a taxa de sucesso da classificação é determinada pela implementação da regras de Bayes,

$$P(W_i|X_a, X_v) = \frac{P(X_a, X_v|W_i) \cdot P(W_i)}{P(X_a, X_v)}, \quad \forall i = 1,...,C \tag{1.7}$$

Como $P(X_a, X_v)$ constante durante o processo de classificação, pode ser obtido a classificação óptima sem o cálculo directo das probabilidades à posteriori,

$$P(X_a, X_v) \cdot P(X|W_c) > P(W_i) \cdot P(X_a, X_v|W_c), \quad \forall i = 1,...,C \quad e \quad i \neq c \tag{1.8}$$

1.5.2 Treino

O treino de um sistema de reconhecimento automático de fala consiste no processo de determinação do valor dos parâmetros livres, cujo conjunto vai ser designado por θ, com base na informação contida num conjunto de exemplos T. Cada exemplo é constituído por duas sucessões de vectores de características, X_a e X_v, e a correspondente sucessão de palavras, W_c, ou seja,

$$T = \left\{ \left(X_a^n, X_v^n, W_c^n \right) \qquad n = 1,..., N_T \right\} \tag{1.9}$$

De acordo com a regra de classificação óptima definida por um classificador Bayesiano, os parâmetros dever ser determinados de maneira a maximizar a probabilidade a posteriori estendida ao conjunto de treino,

$$\theta = \arg\max_{\theta} \prod_{n=1}^{N_T} p_\theta\left(W_c^n \middle| X_a^n, X_v^n\right) \tag{1.10}$$

Para simplificar a complexidade do processo de treino, e de modo a ganhar tempo de processamento, é usual realizar uma aproximação à expressão (1.10). Embora o sistema perca a nível discriminativo, os parâmetros passam a ser determinados de acordo com o critério da maximização de probabilidade conjunta de X_a, X_v e de W_c estendida ao conjunto de treino,

$$\theta = \arg\max_{\theta} \prod_{n=1}^{N_T} p_\theta\left(X_a^n, X_v^n \middle| W_c^n\right) \cdot p_\theta\left(W_c^n\right) \tag{1.11}$$

A utilização dos modelos de Markov não onbserváveis na modelação apresenta a grande vantagem do treino poder dispensar a tarefa árdua de segmentar e anotar a base de dados completa, sendo suficiente iniciar isoladamente os modelos com alguns exemplos de treino. Deste modo, pode ser implementado um processo iterativo de treino, baseado no algoritmo do Viterbi ou de Baum-Welch, que utiliza todos os exemplos disponíveis para treinar embebidamente os modelos.

1.5.3 Classificação

A classificação[1] das duas sucessões de vectores de características, X_a e X_v, consiste no processo de determinação da sucessão de vectores de palavras W_c, correspondente. De acordo com as regras utilizadas para um classificador Bayesiano, X_a e X_v são atribuídos à classe W_c que, entre todas, apresenta o maior valor de $P\left(X_a, X_v \middle| W\right) \cdot P\left(W\right)$.

O problema consiste em pesquisar o espaço das frases possíveis de maneira a encontrar aquela que permite a maior probabilidade conjunta $P\left(X_a, X_v, W\right)$. Embora uma pesquisa exaustiva seja possível em sistemas com vocabulários pequenos, a implementação de uma estrutura integrada permite a utilização de algoritmos bastante eficientes (como por exemplo o de Viterbi).

[1] Conhecida habitualmente por reconhecimento

1.5.4 Unidades básicas, Modelação e Gramáticas

A selecção correcta das unidades básicas é um factor fundamental no desenvolvimento de um sistema de reconhecimento. Nos sistemas de reconhecimento de fala, a dimensão do vocabulário pode ser decisivo na sua complexidade. Quanto maior for a dimensão do vocabulário maior será a possibilidade de confusão entre os elementos que o constituem, embora, nem sempre seja assim, uma vez que depende pricipalmente dos elementos que compõem o vocabulário.

É usual utilizar como unidade básica a menor unidade possível que se pode retirar do sinal a classificar, no entanto, esta decisão deverá passar pelo tipo de sistema que se está a implementar. Por exemplo, num sistema de pequeno vocabulário é perfeitamente aceitável que se utilize um só modelo para cada palavra, caso o conjunto de treino o permita. Nos sistemas de grande vocabulário é usual utilizar unidades mais pequenas mas modeladas com contexto de modo a suprir a perda de precisão que ocorre. Essa perda deve-se às várias unidades poderem aparecer em diferentes palavras, com contextos diferentes, o que leva a apresentarem características mais difusas.

O objectivo da modelação é construir um modelo capaz de determinar com bastante precisão as probabilidades de observar uma qualquer sucessão de vectores de características condicionada ao conhecimento da frase. É importante referir que essa modelação global da frase está extremamente dependente do contexto linguístico $P(W_c)$, ou seja, das restrições impostas pela gramática.

A qualidade dos modelos é fundamental na classificação final, ou seja, no desempenho do classificador. Os modelos de Markov conseguem de um modo bastante consistente, implementar técnicas com uma estrutura bastante simples que apresentam resultados com grande taxa de sucesso. Estes modelos serão apresentados com mais detalhe na secção 2.6.

Como foi referido, a utilização de restrições linguísticas no processo de reconhecimento simplifica bastante a complexidade da tarefa, permitindo melhorar o desempenho do classificador. O uso de gramáticas permite, sem ser necessário conhecimento do sinal, estabelecer probabilidades de ocorrência das unidades seleccionadas nas frases. O principal objectivo das gramáticas pode ser visto como a redução do número de palavras possíveis[1] que podem acontecer em cada momento de decisão. Deste modo, um sistema pode ser visto como uma árvore em que cada nó pode sair

[1] Normalmente designado por vocabulário activo

um determinado conjunto de ramos. As gramáticas estão por trás de toda essa estrutura, limitando as ramificações possíveis em cada nó. Um modo de avaliar a complexidade de uma determinada tarefa de reconhecimento é através da complexidade da árvore que implementa a estrutura do sistema, e que se traduz num valor que representa a sua perplexidade. As gramáticas baixam a complexidade do sistema, o que se traduz numa redução da perplexidade associada.

1.6 História do Reconhecimento Automático de Fala

A evolução do reconhecimento de fala caminhou paralela à das engenharias em especial das relativas às ciências da fala.

Os primeiros estudos associados surgiram no final do século XIX, e com o intuito inicial de vencer as barreiras de comunicação para as pessoas com problemas auditivos. No entanto o reconhecimento de fala apenas teve um impulso definitivo no início do século XX.

1.6.1 Perspectiva histórica

Após os primeiros testes sem sucesso, no início do século XX, na década de 1920, surgiu a primeira máquina com capacidade de reconhecer fala. Esta máquina, conhecida por Rádio Rex [45], não era mais que um boneco comercial (um cão celulóide) e embora com óbvias limitações na área do reconhecimento de fala, foi universalmente aceite. Este brinquedo respondia à palavra Rex, activando-o quando esta fosse pronunciada. A técnica de reconhecimento baseava-se na energia contida na vogal 'e' da palavra Rex. No entanto, o brinquedo respondia a muitas mais palavras semelhantes, ou mesmo outros sons, que tivessem uma energia igual ou superior à da vogal 'e'.

Na década de 1940, surgiram os primeiros passos académicos no reconhecimento de fala. O Departamento de Defesa dos Estados Unidos da América financiou um projecto com o objectivo de interceptar mensagens Russas. No final desta década, Chomsky começou a considerar máquinas de estados finitos para caracterizar uma gramática. Logo de seguida considerou que uma língua poderia ser gerada através de gramáticas de estados finitos [46]. Estas considerações deram origem ao campo da Teoria Formal da Linguagem. Esta teoria usa os conceitos algébricos

e a teoria de conjuntos para definir línguas formais como sequências de símbolos. No final desta década iniciou-se o desenvolvimento de algoritmos probabilísticos.

Na década de 1950 deu-se o salto definitivo no reconhecimento de fala. Em 1952, nos laboratórios Bell, surgiu o primeiro reconhecedor propriamente dito. Era um reconhecedor de dígitos dependente do orador. O sistema media uma simples função do espectro energético no tempo, em duas bandas, correspondentes às duas primeiras ressonâncias (formantes) do tracto vocal [47]. Embora seja uma abordagem e análise elementar, foi mais robusta relativamente à variabilidade da fala, que a maioria das técnicas do espectro da fala que surgiram posteriormente.

Em 1958, Dudley introduziu por primeira vez um classificador capaz de avaliar o espectro de modo contínuo. Esta técnica foi completamente inovadora relativamente às aproximações por formantes anteriormente utilizadas [47], tornando-se a nova referência para os desenvolvimentos que se seguiram.

No final da década de 1950 verificavam-se duas vertentes bem distintas, a simbólica e a estocástica [45][48][49]. Do trabalho de Chomsky, realizado sobre a teoria formal da língua e síntese generativa, surgiu a vertente simbólica. Por outro lado, a vertente estocástica desenvolveu-se graças à Engenharia Electrotécnica e à Estatística.

Nas décadas seguintes diversas técnicas foram desenvolvidas e novos sistemas foram surgindo. Martin em 1964 desenvolveu as redes neuronais artificiais (ANN), para o reconhecimento de fonemas e Widrow treinou ANNs para reconhecer dígitos [50]. Em paralelo foram desenvolvidos novos métodos de comparação de padrões de sequências, das quais teve realce uma abordagem determinística, o Alinhamento Temporal Dinâmico (*Dynamic Time Warp* - DTW) e uma abordagem estatística, os Modelos Escondidos de Markov (*Hidden Markov Models* - HMM) [47].

1.6.2 A consolidação e o surgimento do audio-visual

Na década de 1970 surgiu a primeira aplicação comercial de reconhecimento de fala, o VIP 100, que foi introduzida pela *Threshold Technology* [46]. Este sistema de reconhecimento lidava com um vocabulário de pequena dimensão e era dependente do orador.

A década de 1980 foi a da consolidação das técnicas existentes, focadas em problemas mais complexos e exigentes. Novas técnicas *front-end*, para extracção de características foram desenvolvidas, no entanto, a nível estrutural os reconhecedores não sofreram grandes modificações, apenas foram utilizados mais dados para treinar e testar os sistemas em tarefas mais difíceis. Foi nesta altura que houve um grande esforço no sentido de desenvolvimento de dados de investigação (bases de dados) [47].

A meio da década de 1980 surgiu o primeiro sistema de reconhecimento automático através da leitura de lábios [51]. Este sistema transformava a imagem da face do orador numa imagem binária, através de um simples *threshold*, retirando de seguida diversas características visuais da boca como altura, largura, área e perímetro. De seguida essas características eram colocadas num reconhecedor *single-stream*[1] visual, baseado no algoritmo do DTW de modo a recalcular a melhor de duas saídas do sistema base *single-stream*.

Uma das conclusões que se salientavam no final da década de 1980 era a falta de robustez acústicas dos reconhecedores. Esta era a maior preocupação e o principal objectivo traçado no início da década de 1990 era ultrapassar essa debilidade.

A este nível os esforços foram direccionados para o canal de comunicação (microfones), com o objectivo de uniformizar os resultados independentemente das características do microfone utilizado para captação do sinal de fala e ruído acústico.

Outros esforços houve no sentido de melhorar os reconhecedores. Destacam-se a modelação multi-pronuncia, etc. Tarefas como a identificação da língua e a identificação e/ou verificação do orador foram também preocupação e motivo de estudo desta década.

Em relação ao reconhecimento utilizando características visuais, depois dos excelentes resultados obtidos por Petjan [51] muitos estudos se efectuaram na década de 1990, tendo sido prestada bastante atenção à aquisição de bases de dados de suporte assim como ao estudo de algoritmos de detecção das ROI[2] assim como da extracção dos parâmetros visuais.

[1] *Single-stream* – baseado em apenas uma sequência ou série organizada de dados
[2] ROI – *Region Of Interest*

1.6.3 O reconhecimento de fala audio-visual

Como referido anteriormente, a meio da década de 1980 surgiu o primeiro sistema de reconhecimento automático através da leitura de lábios. O método desenvolvido por Petajan foi um passo em frente no RAF para o reconhecimento dependente do orador (um só orador), numa tarefa de reconhecimento isolado de palavras para um vocabulário de dimensão 100 que incluía letras e dígitos.

Desde o primeiro sistema de reconhecimento com processamento de características visuais proposto por Petajan têm surgido inúmeras contribuições [52] a relatar desenvolvimentos na área AVSR[1], principalmente na última década. Os sistemas que têm sido apresentados diferem em três aspectos principais [53]: a análise visual implementada[2], a estratégia de integração audio-visual e o método de reconhecimento utilizado. Devido à escassez de bases de dados audio-visuais, e à dificuldade na sua disponibilização e transmissão, os diversos sistemas não têm sido testados pela mesma base de dados, não havendo muitos estudos comparativos entre eles. Como veremos mais à frente, devido às dificuldades inerentes à recolha, tratamento e armazenamento de uma base de dados audio-visual, principalmente se esta for de grande dimensão, os desenvolvimentos têm sido efectuados sobre bases de dados de pequena dimensão, para poucos oradores e com tarefas de reconhecimento para um pequeno vocabulário [53][54].

Os sistemas desenvolvidos visam tarefas de reconhecimento de palavras sem contexto[3], palavras isoladas [51], dígitos ligados [56][57], letras [56], ou conjuntos de frases de vocabulários para aplicações específicas [58], a maior parte delas em Inglês, mas também existem estudos em Francês [57][59], Alemão[59], Japonês [59], entre outras.

Recentemente foram relatados desenvolvimentos significativos para reconhecimento automático de fala contínua para sistemas de grande vocabulário[4] (LVCSR) [60], assim como em sistemas em que o sinal acústico é degradado devido a problemas de articulação de voz[5] [61] ou efeito de Lombard [62]. O principal problema da utilização das características visuais nos sistemas de reconhecimento automático de fala reside na qualidade do vídeo capturado e seu tratamento, ou seja, no custo computacional associado para o tornarem utilizável em aplicações em tempo útil.

---

[1] *Audio-Video speech recognition* ou *Audio-Visual Speech Recognition*
[2] *Visual front end design*
[3] *Non-sense words*
[4] *Large vocabulary continuous speech recognition*
[5] *Speech impairment*

Já existem estudos que demonstram que a utilização destes sistemas em ambientes específicos é uma possível solução para garantir a robustez do sistema de reconhecimento.

1.6.4 O estado da arte e os novos desafios

Actualmente o reconhecimento automático da fala já faz parte da vida de muitas pessoas, através de um conjunto de aplicações disponível a todos. Algumas destas aplicações são integradas no computador pessoal munindo-os com capacidade de reconhecimento de fala contínua, permitindo, por exemplo, realizar a transcrição para texto escrito de texto falado [63]. Estes sistemas, dependentes do orador em geral, precisam de um treino inicial para adaptação ao utilizador. Com esta metodologia os sistemas ficam orientados para um só orador (o utilizador que o treinou), obtendo-se taxas de reconhecimento bastante elevadas para este orador, mas resultados inferiores com outros oradores que utilizem o sistema. Um sistema referência é o *FreeSpeech* da Philips.

Aplicações de reconhecimento automático de voz incorporadas em telemóveis permitem fazer a marcação automática de números [64]. Outras para reserva e compra de bilhetes de cinema [64] nos Estados Unidos da América têm sido bastante utilizadas. Outros exemplos são os sistemas de Assistentes Interactivos [65] que permitem, por exemplo, obter um número de telefone.

Os reconhecedores automáticos de fala contínua actuais apresentam bons resultados com léxicos até cerca de 60000 palavras para sistemas independentes do orador.

Nos últimos 20 anos as taxas de reconhecimento tiveram uma subida bastante considerável, aproximadamente 30%. Não deve ser esquecido que, para sistemas com vocabulários de grande dimensão, o uso de gramáticas (2-gram, 3-gram e de ordem superior) em conjunto com os modelos linguísticos é responsável pelos resultados actuais.

O reconhecimento de fala audio-visual é uma área muito recente, tendo-se verificado um crescimento exponencial do esforço de investigação. Em apenas duas décadas foi desenvolvido um sistema de reconhecimento de fala contínua para um grande vocabulário. No entanto, e devido às dificuldades inerentes à recolha e processamento da componente visual, o estado de desenvolvimento está longe do verificado nos sistemas de reconhecimento de fala tradicionais. Assim que forem superadas as barreiras tecnológicas associadas e a recolha e processamento de imagem em tempo real passem a ser uma realidade, o processamento audio-visual poderá

tornar-se num dos principais métodos para proporcionar aos sistemas de reconhecimento de fala a robustez necessária a um desempenho excelente em ambientes acusticamente desfavoráveis.

Para o Português Europeu, o desenvolvimento de sistemas de reconhecimento automático de fala remonta principalmente à última década. Desde os sistemas mais simples de reconhecimento automático de palavras isoladas, ANTÍGONA, até ao reconhecimento automático de fala contínua para vocabulários de grande dimensão [66], passando pelos mais diversos estudos de técnicas de robustecimento, tem-se verificado uma aproximação gradual ao estado de desenvolvimento que se verifica noutras línguas. No entanto, no que diz respeito ao reconhecimento automático de fala com processamento de características acústicas e visuais, só existe um sistema desenvolvido [67]. Esse sistema projecta um reconhecedor automático de fala contínua, para um vocabulário de pequena dimensão, que consiste numa agenda de números telefónicos. A dimensão do vocabulário é de 38 e é constituída por dígitos e caracteres alfabéticos. Este foi o primeiro sistema a surgir para o Português Europeu tendo sido gravada uma base de dados audio-visual [67] com uma qualidade excelente. A taxa de acerto na componte visual que o sistema apresentou foi 67%. Fica sem dúvida a noção que no reconhecimento audio-visual, o estado de desenvolvimento para o Português Europeu ainda está numa fase inicial comparativamente com o que se verifica para outras línguas.

Apesar dos bons resultados que se tem atingido, é sentimento geral que ainda existe um longo caminho a percorrer, sendo esse caminho ainda mais longo no desenvolvimento de sistemas audio-visuais.

1.7 Motivações e a Aplicação

A tecnologia actual tem vindo a sofrer grandes avanços a nível da interacção homem-máquina. Nos computadores, a interacção homem-máquina tradicional, realizada através do teclado e rato, tem vindo a ser substituída por modos mais naturais ao homem, como são o toque, os gestos e a voz. Estas transformações que se encontram actualmente em desenvolvimento, necessitam de muita evolução, e muito esforço tanto a nível de investigação, como esforço tecnológico, de modo a num futuro próximo as interfaces homem-máquina (HCI) se tornarem mais fáceis de usar, transparentes e robustas, independentemente do ambiente de utilização. Neste sentido,

não é apenas importante melhorar o reconhecimento e compreensão dos diferentes modos[1] nos diversos ambientes, mas também desenvolver arquitecturas e tecnologias de modo a escolher o modo, ou modos, mais adequado ao contexto e conteúdo da interacção. Pode-se então afirmar, que a cognição humana poderá ser um bom caminho para o desenvolvimento futuro das HCI, desde que se desenvolvam capacidades de percepção, integração e interpretação dos modos auditivo, visual, gestual e por toque.

Nas últimas décadas tem-se verificado desenvolvimentos significativos nos diferentes modos de HCI. Por exemplo, no áudio através do reconhecimento de fala e do orador, no vídeo através da localização da face e identificação da pessoa. No entanto, tem sido cada vez maior o interesse no estudo da capacidade humana para processar áudio e vídeo em simultâneo, de modo a melhorar o reconhecimento da fala, identificação do orador, etc., em condições ambientais reais. A utilização do aúdio e vídeo em simultâneo tem permitido desenvolvimentos em áreas críticas das HCI, nomeadamente no reconhecimento da fala e do orador, separação das fontes em ambientes multi-orador e localização da fonte, entre outras.

Um ponto importante da utilização em simultâneo das componentes áudio e visual nas HCI, é que as características da componente visual extraídas da face do orador, embora correlacionadas com as características acústicas do sinal de fala associado, complementam as retiradas da componente áudio que tradicionalmente se utilizam.

Por outro lado, uma vez que a discriminação visual não se degrada com o aumento do ruído acústico, faz todo o sentido que esta abordagem seja alvo de estudo.

O interesse e mais valia da tecnologia multi-modal já está experimentalmente demonstrado em algumas línguas, mas, no caso particular do Português Europeu, ainda existe apenas um estudo, tal como foi referido na secção anterior (ver secção 1.6.4).

Além de ser importante tentar confirmar as expectativas favoráveis, é também muito importante estimar se a eficácia desta abordagem é ou não relevante. Chega ter presente as diferenças acentuadas, entre as várias línguas, que caracterizam a expressão visual do processo de produção da fala, para compreender que não é legítima a simples extrapolação de resultados de uma língua para outra qualquer. Por exemplo, os conjuntos de visemas definidos nas várias línguas apresentam diferenças significativas.

[1] Gestos, voz, etc.

Um outro factor que motivou este trabalho foi o de colocar a ciência e a tecnologia ao serviço das pessoas com deficiências. Ultrapassar as barreiras com que se deparam as pessoas com deficiências no dia a dia é cada vez mais uma das preocupações actuais, reflectindo-se em desenvolvimentos nas mais variadas áreas. Examinando os produtos existentes para pessoas com deficiência físicas, chegamos à conclusão que o reconhecimento de fala neste domínio pode ser de vital importância.

Decidido o âmbito tecnológico (domínio audio-visual) e a orientação social (pessoas com deficiências), foi necessário especificar o sistema que se iria implementar. Além disso, pretendia-se iniciar o desenvolvimento de um sistema para o Português Europeu. Desde logo surgiram dois contratempos. Por um lado, apenas foi encontrada uma base de dados audio-visual para o Português Europeu [67], e por outro lado, não foi encontrada, para qualquer língua, nenhuma base de dados audio-visual para pessoas com deficiências. Estes factos levantaram imediatamente a necessidade de criar uma base de dados audio-visual de raíz e encontrar um orador para a gravar. Ao ser necessário criar uma base de dados de raíz, e devido às restrições temporais para realização desta tarefa, o sistema a desenvolver teve que ser um sistema de reconhecimento pequeno vocabulário. Depois de várias hipóteses, ficou decidido que o sistema iria reportar à implementação das funções de comando de uma calculadora científica, uma vez que com um pequeno vocabulário permitiria implementar inúmeras funções. Para gravar a base de dados foi convidado um estudante da Faculdade de Engenharia da Universidade do Porto, portador de distrofia muscular[1]. Apesar de já existirem muitas normas técnicas para facilitar a integração dos cidadãos com mobilidade limitada, assim como desenvolvimentos tecnológicos noutras áreas, na área do reconhecimento da fala esses sistemas são praticamente inexistentes. O interesse no desenvolvimento deste sistema ficou mais reforçado pela doença do orador escolhido.

De um modo geral, pode-se assumir que, do ponto de vista tecnológico, a aproximação audio-visual para desenvolvimento dos reconhecedores de fala para pessoas com distrofia muscular, apresentam duas vantagens óbvias e essenciais:

- Geralmente o sinal acústico é fraco, portanto, o robustecimento acústico é um problema crítico;

[1] Doença neuro-muscular, caracterizada por um enfraquecimento muscular que afecta não só os membros superiores, mas também influencia a produção da fala, em menor grau.

- Em muitas aplicações o utilizador mantém uma posição estável, logo, podem ser extraídas características visuais robustas.

Para uma melhor compreensão das limitações do orador, é realizada na secção 1.7.1 uma breve descrição sobre as doenças neuromusculares.

É importante que fique bem claro, pelos motivos que já aqui foram referidos, que este trabalho visa iniciar um estudo nesta área e não obter um sistema de reconhecimento automático de fala contínua totalmente implementado para funcionar em tempo real. Isto deve-se à limitação temporal para a realização deste trabalho, quer pela complexidade dos assuntos abordados (tratamento audio-visual), quer pela inexistência de material necessário ao desenvolvimento do sistema (foi necessário criar uma base de dados de raíz). Este trabalho pretende pois implementar os motores necessários ao desenvolvimento de um sistema de reconhecimento automático de fala audio-visual, optimizá-lo e demonstrar as vantagens da utilização da componente visual em simultâneo com a componente acústica. A interface entre o utilizador e os motores ficou para segundo plano.

1.7.1 As Doenças Neuromusculares

A descrição mais completa e precisa da Distrofia Muscular foi feita por Guillaume-Benjamin-Amand Duchenne (1806-75), em 1868. Desde então a doença passa a ser conhecida como Distrofia pseudo hipertrófica ou Distrofia Muscular de Duchenne (DMD).

Antes, outros médicos haviam descrito a doença entre eles o cirurgião escocês Charles Bell. O médico inglês Edward Meryon descreveu os achados microscópicos dos meninos afectados em exames pós-mortem.

Em 1858, Duchenne documentou o caso de um menino de 9 anos que perdeu a capacidade de andar devido a uma doença muscular. Em 1868 publicou 13 casos e fez inúmeras observações importantes em relação a sinais e sintomas e ao facto de que a deterioração intelectual pode fazer parte da clínica da doença. Também através de suas observações concluiu-se que a patologia era transmitida por herança, afectando principalmente meninos. De forma errada, Duchenne sempre acreditou que a doença decorria de alterações no Sistema Nervoso.
Ernest Leyden sugere que as distrofias herdadas devem ser classificadas em categorias separadas daquelas consequentes à lesão dos nervos.

William Erb foi o primeiro a tentar diferenciar os vários tipos de distrofias, classificando-as segundo a idade de início. Erb teorizava, erroneamente, que a DMD era causada por nutrição inadequada.

Em 1879, o neurologista inglês William Gowers descreveu o modo como os meninos afectados pela DMD se tentavam levantar. Esta manobra passou a ser conhecida como "sinal de Gowers".
Nos anos 50 houve importantes progressos, incluindo a fundação da Associação de Distrofia Muscular e um estabelecimento de uma classificação mais fidedigna da distrofia muscular. Esta classificação foi alterada em 1957 por P.E. Becker, que descreveu uma variante menos severa de DMD e que levou o seu nome.

Em 1986, com a técnica de DNA recombinante descobre-se que um gene, quando defeituoso, causa a Distrofia Muscular de Duchenne e Becker. Em 1987 é identificada a ausência ou diminuição de uma proteína, denominada distrofina, nos meninos afectados.

Dois trabalhos recentes reforçaram a importância do estudo de Edward Meryon (1807-80) que, dez anos antes de Duchenne, descreveu em 1851 numa reunião da Sociedade Real de Medicina e Cirurgia (publicada em 1852) nove casos de distrofia muscular em três famílias. Além de descrever o carácter familiar da doença, que não foi descrito por Duchenne, relatou que a doença afecta meninos e que afecta os músculos, sem encontrar alterações em nervos da coluna e gângleos.

Da Distrofia Muscular Progressiva (DMP) podemos então referir que engloba um grupo de doenças genéticas, que se caracterizam por uma degeneração progressiva do tecido muscular e nervos periféricos.

Até ao presente momento tem-se o conhecimento de mais de trinta formas diferentes de DMPs, algumas mais benignas e outras mais graves, que podem atingir crianças e adultos de ambos os sexos. Todas atacam a musculatura, mas os músculos atingidos podem ser diferentes de acordo com o tipo de DMP. No entanto todas elas apresentam características semelhantes:

- Falta de forças;
- Por vezes atrofia dos músculos;
- Podem atingir qualquer grupo muscular e surgir em qualquer idade;
- Pode também estar afectada a sensibilidade à dor e à temperatura;

- Grande parte são doenças congénitas (doenças familiares);
- Nalguns casos, como na Miastenia, a falta de forças varia ao longo do dia e pode mesmo normalizar com repouso.

De entre um conjunto de conselhos que existem para doentes Neuromusculares de modo a orientá-los no seu dia a dia, é de realçar que o âmbito deste trabalho se enquadra num desses conselhos, que o valoriza do ponto de vista social.

"Adapta, dentro do possível, os locais onde te encontras com ajudas técnicas que substituam os músculos afectados."

CAPÍTULO 2

Sistema Audio-Visual de Reconhecimento de Fala Contínua

Neste capítulo é apresentado o sistema de reconhecimento automático de fala contínua. Inicialmente é realizada uma caracterização do sistema. Seguidamente é realizada uma abordagem às metodologias seguidas na sua implementação e às principais considerações adoptadas no seu desenvolvimento. Finalmente é abordado o problema do reconhecimento automático de fala contínua, baseado no processamento simultâneo das características acústicas e visuais.

2.1 Considerações gerais

O presente trabalho teve como principal objectivo a construção de um sistema audio-visual de reconhecimento automático de fala contínua, dependente do orador e com um vocabulário de pequena dimensão, de modo a poder verificar em que situação pode ser benéfica a abordagem audio-visual num sistema de reconhecimento automático de fala.

Um dos principais conhecimentos a adquirir com este trabalho refere-se às técnicas mais comuns em sistemas de reconhecimento automático de fala *standard* e sua introdução na abordagem *multi-stream*.

Dadas as características e as vantagens, amplamente demonstradas na aplicação de sistemas de reconhecimento de fala, foram utilizados os modelos de Markov não observáveis (HMMs) na modelação das unidades acústicas elementares, que no sistema implementado são ao nível da palavra.

O código fonte dos programas utilizados foi realizado, ou adaptado de sistemas disponíveis desenvolvidos "de raiz", sem recorrer a ferramentas dedicadas ao reconhecimento de fala (como por exemplo o HTK Toolkit). Os ambientes de desenvolvimento foram o compilador Visual Studio C/C++ e o software MATLAB.

2.2 Tarefa de reconhecimento

O sistema audio-visual de reconhecimento automático de fala contínua foi desenvolvido para uma aplicação específica (interface de uma máquina de calcular científica), com um vocabulário de pequena dimensão, para um só orador.

A tarefa de reconhecimento é definida em função das especificações da aplicação final. Para o sistema desenvolvido, podemos apresentar a tarefa de reconhecimento do seguinte modo:

Seja $x_a(t)$ o sinal acústico e $x_v(t)$ o sinal visual de um sinal de fala que se pretende reconhecer. Após pré-processamento, esse sinal pode ser representado no domínio discreto por duas sequências de vectores,

$$x_a(k), \quad k = 1, \quad 2, \quad 3, \quad \ldots \tag{2.1}$$

$$x_v(k), \quad k = 1, \quad 2, \quad 3, \quad \ldots \tag{2.2}$$

correspondentes a uma sucessão de palavras,

$$W_1^k = \{w_1, w_2, w_3, \ldots, w_k\}, \quad w_i \in \Gamma_w \tag{2.3}$$

válida nesta tarefa.

O objectivo da tarefa de reconhecimento é determinar a sucessão correcta de palavras produzidas pelo orador que corresponde ao sinal audio-visual analisado,

$$x_a(k), x_v(k) \longrightarrow W_1^K \tag{2.4}$$

Na tarefa de classificação devem ser utilizados dois tipos de modelos:

- O modelo audio-visual, M_{av}, associado à realização audio-visual das frases nesta tarefa de reconhecimento. Este tipo de modelo requer uma fase inicial de treino por parte do sistema, que é realizado através da base de dados específica da aplicação. Esta base de dados resultou da leitura natural de um conjunto de frases específicas (transcrição ortográfica das mesmas), num ambiente acusticamente favorável, com ausência de ruídos exteriores significativos.

- O modelo linguístico, L, definido especificamente para a tarefa. Este modelo pode ser dependente das regras sintácticas do domínio da língua (ou da aplicação), ou pode ser criado a partir das realizações da base de dados (sub-conjunto utilizado para treino). No primeiro caso é utilizada uma gramática *Word Pair*, que estabelece as palavras que podem suceder a uma qualquer pertencente também ao vocabulário, Γ_{ω}, com 68 palavras. No segundo caso, o modelo é representado através de uma *Bigram* que estabelece a probabilidade de todas as realizações de pares de palavras encontradas no conjunto de treino da base de dados.

Deste modo, a tarefa de reconhecimento vem,

$$x_a(k), x_v(k) \xrightarrow[M_{av}, L]{} W_1^K \tag{2.5}$$

2.3 Base de Dados

Para o desenvolvimento do sistema foi necessário criar de raiz uma nova base de dados audio-visual. Os motivos que levaram a esta necessidade, prendem-se, por um lado à especificidade do sistema que se decidiu implementar e, por outro lado, à escassez de recursos disponíveis neste campo específico. Sendo apenas conhecida uma base de dados audio-visual para o Português Europeu [67], foi decisiva a necessidade de criar uma nova base de dados de raiz.

Uma desvantagem óbvia associada à criação de uma nova base de dados reside no facto desta ser completamente desconhecida, não existindo referências a nível de taxas de reconhecimento. O facto de não haver tempo suficiente para ser validada pode também ser um obstáculo à rapidez de resolução de problemas que inevitavelmente surgem no desenvolvimento do sistema, e que são características próprias e intrínsecas das condições de gravação (orador, material e condições de gravação, etc.).

É importante referir que criar a base de dados de raiz traz benefícios óbvios. Todas as frases são geradas para uma aplicação pré-definida, ou seja, as frases são dedicadas à aplicação projectada.

2.4 Estrutura do Reconhecedor

A estrutura de reconhecedor automático de fala *standard*, em que só existe um sinal a classificar, apresenta dois módulos fundamentais, o módulo de análise e o de classificação. Na abordagem audio-visual a estrutura do reconhecedor deverá estar preparada para receber e processar dois sinais, acústico e visual, de modo a realizar uma classificação final.

Têm sido exploradas várias abordagens [68][55][69][70][71][72][73][74][75] na integração audio-visual em sistemas de reconhecimento, que vão desde o processamento individual de cada sinal (desenvolvimento de classificadores *single-stream*), implementando um método final de classificação, até sistemas que combinam os dois sinais logo após a fase de análise. Essas abordagens diferem principalmente a nível da estrutura implementada e da terminologia seguida. O objectivo destas abordagens consiste em desenvolver um classificador bimodal que apresente um desempenho superior aos classificadores *single-stream* (áudio e vídeo).

As técnicas de integração audio-visual tem sido agrupadas em métodos *feature fusio*n ou *decision fusion*, conforme a zona do reconhecedor onde a combinação audio-visual é realizada.

Nas técnicas *feature fusion*, como está apresentado na Figura 2.1, os dois sinais são combinados no bloco de análise, logo após a extracção das suas características, resultando apenas um vector de características à saída do módulo de análise. Deste modo, estas técnicas apenas necessitam de um classificador, tal como na classificação *single-stream*. A combinação dos *streams* dos dois sinais pode ser realizada por simples concatenação ou através de uma transformação dos dois streams [55][71][76].

As técnicas de *feature fusion* mais utilizadas são a *plain feature concatenation* [55], *feature weihting*[1] [71][73], e *hierarchical linear discriminant feature extration* [76]. Outras técnicas

[1] Mais conhecida por *direct identification fusion*

como motor *recording*[1] fusion [71] e *audio feature enhacement* [77][78] e *concatenative feature fusion* [55] também pertencem a este grupo.

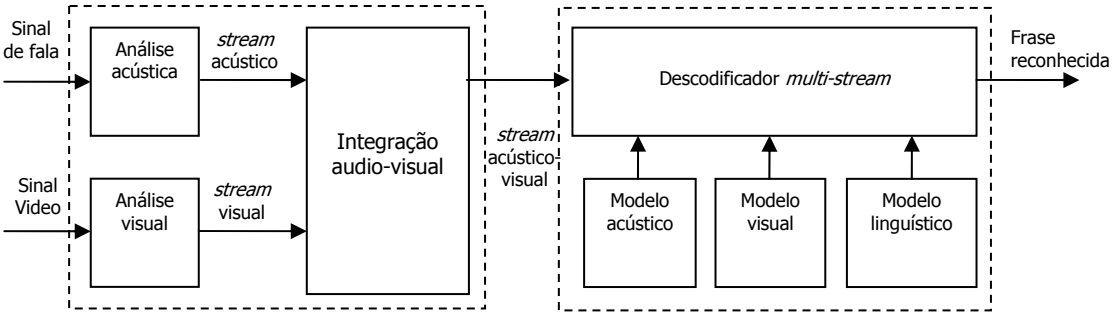


Figura 2.1 - Estrutura do reconhecedor *multi-stream* (abordagem *feature fusion*)

Nas técnicas *decision fusion*, como está apresentado na Figura 2.2, cada um dos sinais, após sair do módulo de análise, é processado por um classificador diferente (é necessário implementar dois classificadores *single-stream*), sendo posteriormente realizada uma classificação final.

Uma das técnicas *decision fusion* mais utilizada consiste em utilizar vários classificadores em paralelo estabelecendo posteriormente um ranking das classes [53][69][72][75].

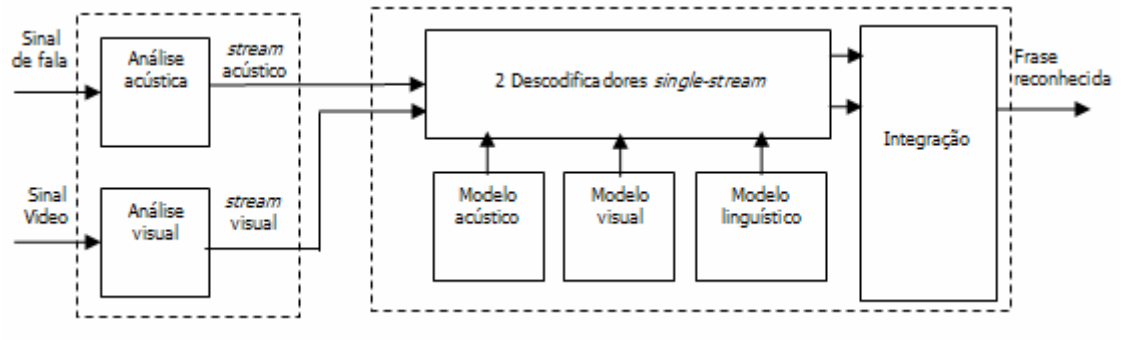


Figura 2.2 - Estrutura do reconhecedor *multi-stream* (abordagem *decision fusion*)

[1] Também conhecida por *dominant motor*

Um caso particular da integração audio-visual ocorre quando a combinação dos dois *streams* é realizada durante o processo de classificação. Esta combinação é realizada ao nível da modelação das classes do sistema. Embora diversos autores incluam este tipo de integração no grupo das *decision fusion*, existem abordagens que a classificam num grupo intermédio, *híbrid fusion* (ver Figura 2.3), uma vez que apenas é necessário um classificador. Estas técnicas, também conhecidas *model fusion*, podem ser vistas como uma combinação das características das técnicas *feature* e *decision fusion*.

A implementação mais usual nos métodos híbridos consiste em combinar linearmente no classificador as verosimilhanças calculadas dos dois *stream*, através da atribuição de um peso específico. Esta abordagem também é conhecida por *separate identification*.

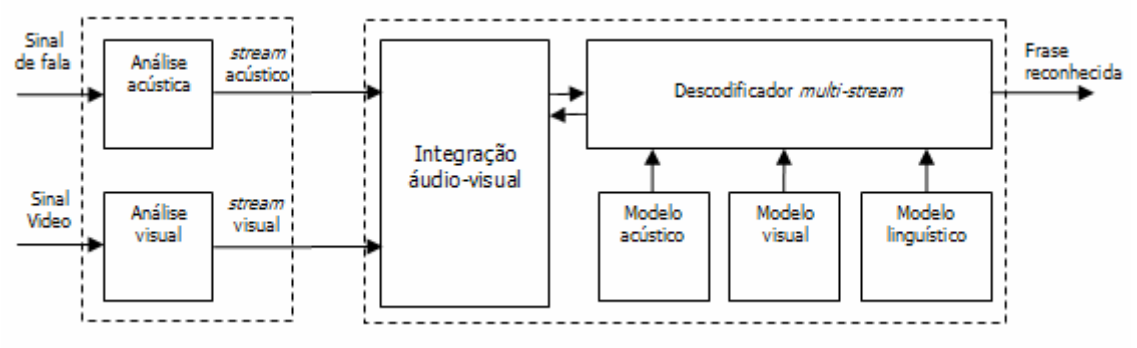


Figura 2.3 - Estrutura do reconhecedor *multi-stream* (abordagem *hibrid fusion*)

Os autores que defendem apenas a classificação das técnicas de integração em *feature fusion* e *decison fusion*, realizam divisão dos últimos em três sub-grupos distintos, de modo a distinguir o tipo de integração que é realizada. Neste sentido, os métodos *decision fusi*on podem ser classificados por *early*, *intermediate* e *late integration*, se a integração for realizada ao nível sub-fonético, de fonema[1] ou ao nível da frase, respectivamente.

Na *early integration*, os streams de características acústicas e visuais são concatenados no mesmo vector de observações. Nesta abordagem é assumido que os *streams*, áudio e vídeo, têm a mesma definição temporal, o que está longe de ser verdade. As técnicas existentes para implementar as técnicas de *early integration* precisam de mais dados para treino, do que treinando os modelos acústicos e visuais separadamente. Deste modo, o espaço de representação cresce significativamente, tornando o processo mais demorado.

[1] Ou palavra

Na abordagem *late integration* os streams acústicos e visuais são processados independentemente, o que geralmente permite maior precisão na modelação espacial e temporal de cada modalidade. Esta abordagem apresenta a vantagem de permitir arquitecturas de classificação independentes para cada um dos streams, sendo no entanto, necessário um terceiro classificador para integra-los. No caso do reconhecimento de fala audio-visual robusto, o processo de integração geralmente analisa o nível de ruído (SNR) de modo a decidir qual o peso específico que deve atribuir a cada *stream*.

Na *intermediate integration* são reunidas características dos dois métodos anteriores. Esta abordagem está claramente representada em [60][79] onde dois canais lineares de unidades neuronais são conectados de modo a que possam interagir entre eles. Stork apresentou resultados que demonstram que esta abordagem pode atingir uma taxa de reconhecimento ao nível dos métodos anteriores [80].

Neste tipo de classificação, os métodos híbridos são englobados na *intermediate integration*.

2.4.1 Estrutura implementada

A estrutura do reconhecedor implementado, apresentada na Figura 2.4, é semelhante à maioria dos sistemas estatísticos utilizados em aplicações de fala contínua e vocabulário de pequena ou média dimensão.

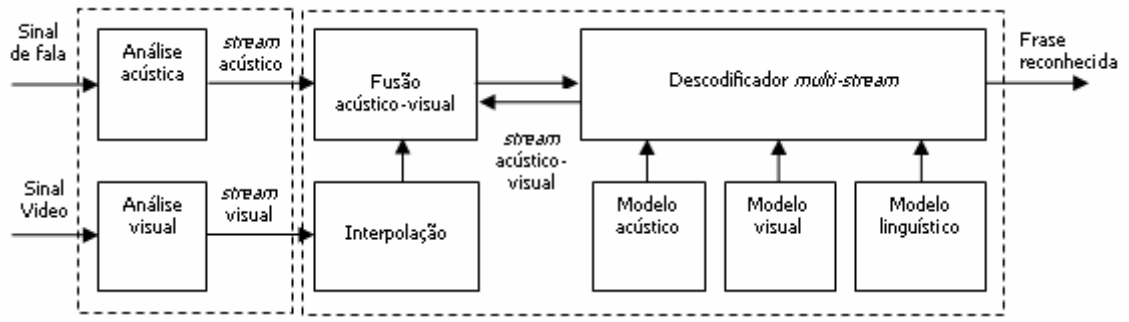


Figura 2.4 - Diagrama de blocos do reconhecedor *multi-stream* implementado

No primeiro módulo, a partir dos sinais, acústico e visuais, é calculada uma sucessão de vectores de características para cada um desses sinais.

O descodificador *multi-stream* consiste numa estrutura, que através de um algoritmo de pesquisa baseada no modelo acústico-visual e linguístico[1], determina uma sucessão de palavras que melhor representam os vectores de características, acústicas e visuais extraídos, segundo o critério definido pelo algoritmo de pesquisa. O modelo acústico-visual, que modelam as unidades elementares de fala (neste caso ao nível da palavra), são formalizados através dos modelos de Markov não observáveis do tipo semi-contínuo, e o modelo linguístico é definido através de gramáticas do tipo *Word Pair* e *N-gram*.

A abordagem de integração audio-visual consiste numa implementação híbrida, onde a fusão é realizada ao nível do modelo da unidade elementar de fala do sistema. O modelo de cada unidade de fala pode ser representado como apresentado da Figura 2.5.

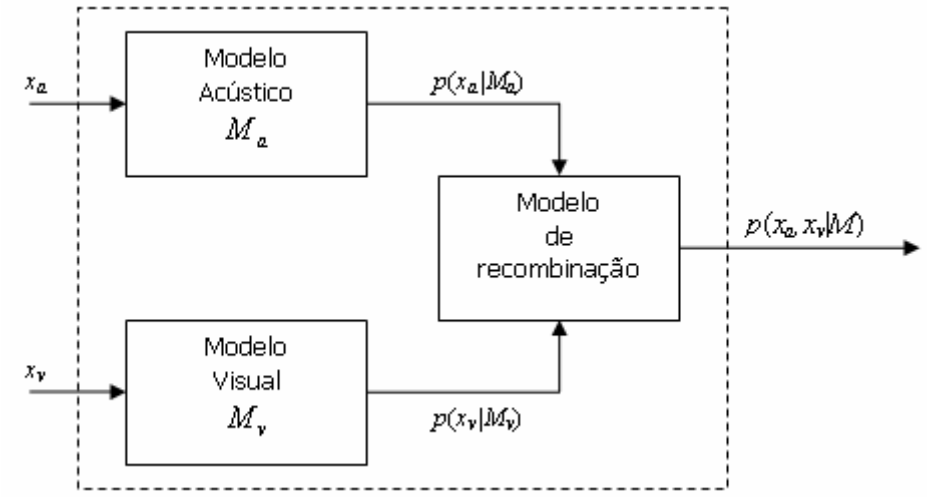


Figura 2.5 - Modelo audio-visual composto de uma unidade de fala M

Da Figura 2.5 observa-se que os streams são processados independentemente até ser obtida a verosimilhança das observações a cada um deles. Estes valores são então integrados no modelo de recombinação, originando o resultado pretendido.

2.5 Módulo de Análise

O módulo de análise ou de pré-processamento consiste num conjunto de operações sobre o sinal acústico e sobre o sinal visual. Esse tipo de operações é específica para cada um dos dois sinais e tem como principal função preservar as características relevantes para o processo de classificação, e eliminar as características redundantes, de modo a minimizar o espaço de

[1] O modelo linguístico é opcional

representação e o tempo de processamento do classificador. Este processo deve ser efectuado sobre todo o material da base de dados para treino, desenvolvimento e teste final. O pré-processamento eficiente dos sinais pode ser fundamental no processo de classificação do sistema.

2.5.1 Pré-Processamento do sinal acústico

O pré-processamento é fundamental para o desempenho correcto de um qualquer reconhecedor. Este divide-se em várias etapas, como são a pré-ênfase, a remoção do ruído às baixas frequências e da componente DC (ver Figura 2.6).

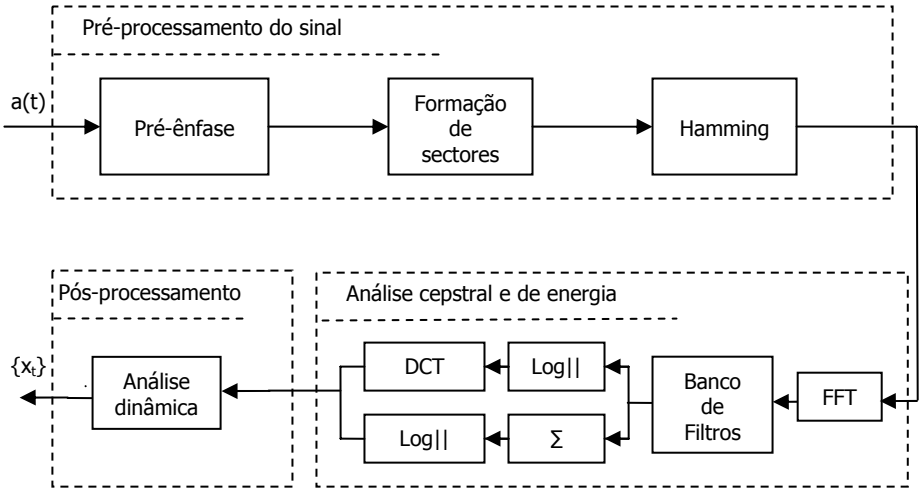


Figura 2.6 - Módulo de extracção de características acústicas

Na parte inicial do módulo de análise, é realizado o designado pré-processamento, onde o sinal é preparado para a fase de extracção das características (energia e coeficientes mel-cepstrais), que consiste na operação de pré-ênfase, seguida da formação de sectores a que vão corresponder os vectores de características.

Na segunda parte, a partir de cada sector do sinal vão ser calculados os respectivos coeficientes mel-cepstrais, com a ajuda de um método baseado num banco de filtros. Por análise cepstral é calculada a energia.

Na última fase, é acrescentada informação local adicional[1], sobre a trajectória dos vectores de características calculados.

[1] É calculada a primeira derivada das características determinadas

À saída do módulo extractor de características, cada sinal acústico vai ser representado por um conjunto de vectores de características definidos num espaço 26-dimensional ($\Re^{26}$).

2.5.1.1 Pré-ênfase do sinal

O som é amostrado a 22,05KHz, 16 bits, mono, e sofre uma filtragem inicial com uma frequência superior de corte de 8Khz (uma vez que o espectro de sinais de fala não supera esse valor) para eliminar a componente DC e atenuar o ruído às baixas frequências e evitar o *aliasing*.

A pré-ênfase consiste na filtragem do sinal de fala com um filtro linear de primeira ordem. O objectivo desta filtragem é a de nivelar a representação espectral do sinal. A função de transferência do referido filtro é dada por:

$$H(z) = 1 - \alpha . z^{-1}, \qquad 0{,}91 < \alpha < 0{,}99 \tag{2.6}$$

Constata-se, que esta operação, quando implementada, melhora significativamente a taxa de sucesso de um reconhecedor. A sua principal acção é um maior aumento da energia nas bandas de maior frequência, onde se encontra mais informação significativa dos sinais, por exemplo, os correspondentes a sons fricativos.

2.5.1.2 Formação de sectores e Janelamento de Hamming

A sucessão de vectores de características é determinada analisando uma sucessão respectiva de sectores do sinal.

Pode-se entender a sucessão de sectores como o resultado de observações do sinal em sucessivos intervalos de tempo, com a ajuda da janela de Hamming (podia ser utilizada outra janela, mas em aplicações deste tipo, é a que apresenta melhores resultados). Assim cada sector vai resultar da observação de intervalos de 10ms, sendo cada observação realizada por uma janela de Hamming com a duração de 25ms e com uma sobreposição de 15ms do sector anterior.

O bloco seguinte tem como objectivo a segmentação do sinal em janelas. O uso de janelas com duração curta traduz-se numa perda de representação dos harmónicos de excitação nos sons vocálicos, bem como de um alisamento do espectro dos sons. Pelo contrário, o uso de janelas com maior duração, proporciona uma boa representação destes harmónicos assim como um

aumento na definição espectral do ruído, mas por outro lado, existe uma perda de resolução temporal.

Os coeficientes h são calculados através da seguinte fórmula, em que N indica o comprimento da janela:

$$h(k+1) = 0{,}54 - 0{,}46.\cos\left(\frac{2.\pi.k}{N-1}\right) \quad k = 0,\ldots,N \tag{2.7}$$

Esta janela tem a vantagem de reduzir o fenómeno de Gibbs [81], segundo o qual a truncagem abrupta do sinal num dos domínios de representação origina oscilações no outro domínio, em relação a uma janela rectangular.

Visto que as amostras do sinal são bastante atenuadas nas extremidades da janela de Hamming, é necessário sobrepor parcialmente janelas consecutivas. Deste modo usamos no nosso sistema janelas de *Hamming* com 15ms de sobreposição e 25ms de duração da própria janela, valores estes tipicamente usados para sistemas de reconhecimento.

Na Figura 2.7 seguinte podemos verificar a divisão do sinal em sectores.

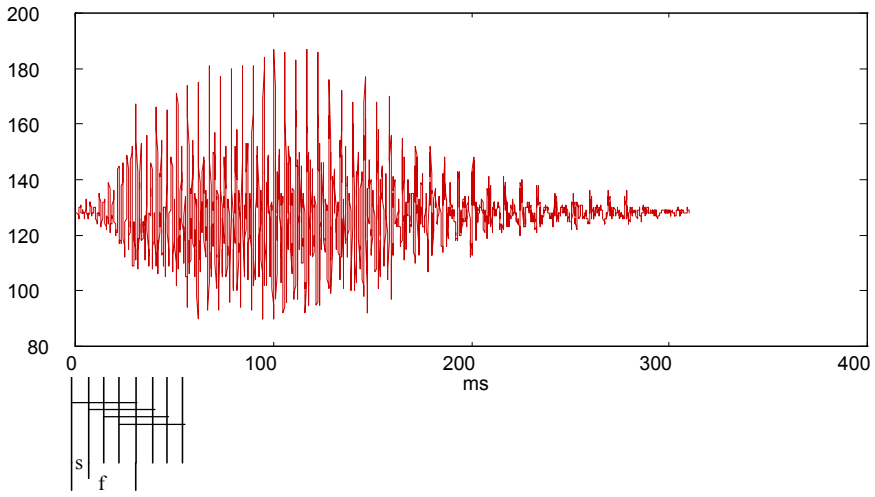


Figura 2.7 - Sectorização do sinal acústico

Uma vez que cada sector tem um comprimento muito pequeno, podemos considerar que o sinal é quase estacionário nesses intervalos e assim podemos aplicar as técnicas de análise espectral de sinal conhecidas para este tipo de sistemas, que são razoavelmente simples.

2.5.1.3 Bancos de Filtros e Cálculo dos Coeficientes Mel-cepstrais

A escolha dos coeficientes mel-cepstrais como parâmetros acústicos nos sistemas de reconhecimento automático da fala teve origem no conhecimento acerca do aparelho auditivo humano e na técnica de análise cepstral.

Mel é uma unidade de medida da percepção do *pitch* ou frequência de tom. Esta unidade não corresponde à frequência linear dos tons uma vez que o ouvido humano não tem uma percepção linear da frequência. Experiências realizadas por Stevens e Volkman levaram ao mapeamento entre a escala da frequência real (em Hz) e a escala da frequência realmente percebida (em Mel). O mapeamento entre estas duas escalas é praticamente linear abaixo de 1KHz e logarítmico acima desse valor [82]. Estudos na área do processamento da fala levaram à confirmação dos benefícios da utilização da escala Mel, ajustada à escala de frequências.

No gráfico da Figura 2.8 é apresentada a correspondência entre a escala linear em Hz e em Mel, cuja conversão é dada por,

$$F_{Mel}(F_{Hz}) = 2595 \cdot \log_{10}\left(1 + \frac{F_{Hz}}{700}\right) \tag{2.8}$$

em que F_{Hz} é a frequência real em Hz e F_{Mel} é a frequência percebida em Mel.

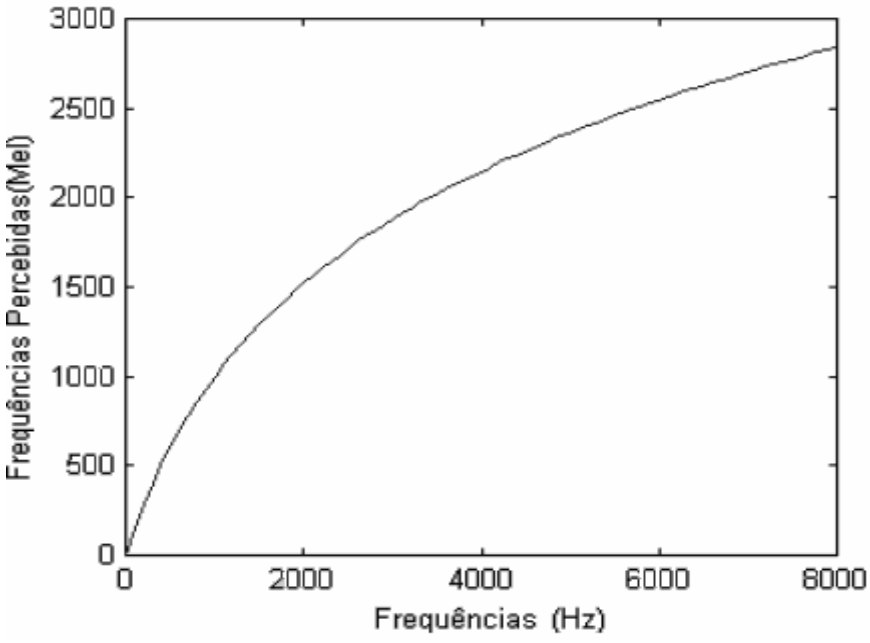


Figura 2.8 - Relação entre a frequência linear em Hz e a frequência percebida em Mel

Os coeficientes mel-cepstrais são utilizados nos sistemas de reconhecimento automático da fala pela sua robustez. Na Figura 2.9 é apresentado o método para determinação dos coeficientes mel-cepstrais.

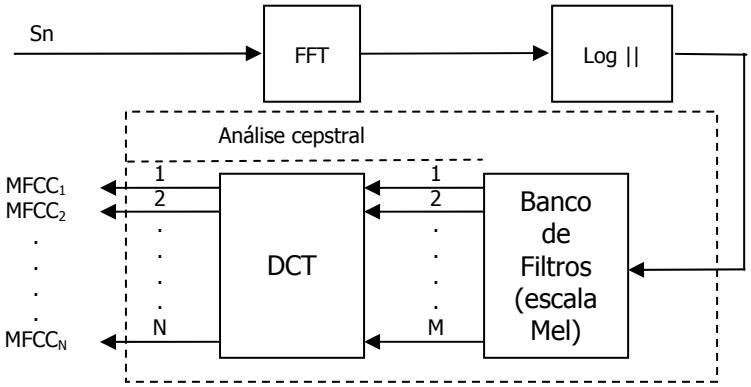


Figura 2.9 - Bloco de extracção dos coeficientes mel-cepstrais

Primeiro obtém-se o espectro de potência de curta duração através de um algoritmo de transformada rápida de Fourier de 512 pontos. De seguida, um conjunto de 24 filtros passa-banda triangulares organizados segundo a escala de Mel é aplicado ao resultado anteriormente obtido. Dos logaritmos das energias deste banco de filtros são extraídos 12 coeficientes mel-cepstrais utilizando a transformada discreta de co-seno (DCT). Esta operação não é mais do que um procedimento heurístico baseado na energia em bandas espectrais espaçadas de acordo com a escala Mel. Em (2.7) apresenta-se a expressão que permite calcular os coeficientes mel-cepstrais.

$$MFCC(i) = \sqrt{\frac{2}{P}} \cdot \sum_{k=1}^{P} m_k \cdot \cos\left(\frac{\pi \cdot i}{P} \cdot (k - 0{,}5) \right) \quad i = 1,2,\ldots,M \tag{2.9}$$

Na Figura 2.10 é apresentado o banco de filtros triangulares às saídas do qual é determinado um sinal e constituem um espectro de potência do sinal. Como se pode verificar, m_k é a energia em cada banda.

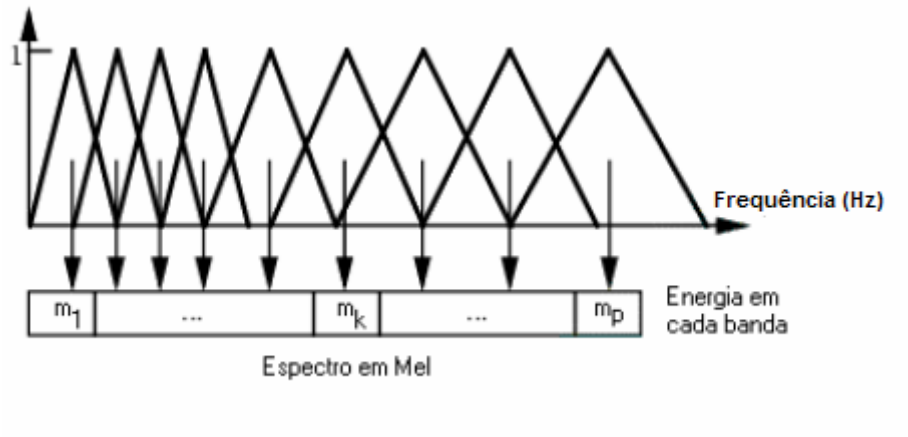


Figura 2.10 - Banco de filtros triangulares na escala Mel

À saída de cada filtro fica disponível uma estimativa da energia do sinal nessa banda.

2.5.1.4 Cálculo da Energia

Um dos elementos do vector de características é o logaritmo de energia do sinal em cada segmento. Este valor é calculado a partir da energia à saída do banco de filtros (2.8),

$$E = \log_{10} \sum_{n=1}^{N} x_n^2 \cdot w_n \tag{2.10}$$

Onde x_n correspondem às amostras do sinal de fala e w_n corresponde a uma janela de *Hamming* de comprimento N.

2.5.1.5 Pós-processamento e características acústicas finais

À entrada do módulo de análise dinâmica foram extraídos 12 MFCCs e um parâmetro representando a energia.

Através dos parâmetros calculados, os modelos acústicos modelam sobretudo as posições instantâneas no estado de representação acústica definidas pelos vector de características, uma vez que cada ponto pode pertencer a várias trajectórias.

Para tornar mais discriminativos os modelos acústicos, estes deveriam modelar a trajectória da sucessão de eventos no espaço definido. O problema é que devido às limitações apresentadas pelos modelos de Markov não observáveis, essa modelação não é realizada satisfatoriamente a não ser que seja inserida informação complementar.

Para resolver este problema têm sido apresentados vários métodos apresentando melhoria de desempenho significativa [83][84]. Essas soluções têm visado, desde processamento dos vectores de características até soluções ao nível do modelo acústico. O primeiro tipo de abordagem visa por exemplo agrupar janelas com contexto e no segundo tipo incluem-se modelos híbridos.

O objectivo pretendido é, de um modo simples e consistente, modelar a trajectória dos vectores de características. O método utilizado consistiu no cálculo de uma regressão linear efectuada sobre os vectores de características nos sectores vizinhos [23].

Segundo este método, para cada um dos parâmetros, $x_n^i =, \quad i = 1,2,\ldots,13$, do vector de características existente, calcula-se um novo termo, designado por "primeira derivada pesada", através da expressão,

$$\Delta x_n^i = \frac{\sum_{k=-2}^{2} k \cdot x_{n+k}^i}{\sum_{k=-2}^{2} k^2}, \quad i = 1,2,\ldots,13 \tag{2.11}$$

No final do módulo de extracção de características acústicas, cada sinal acústico vai ser representado por um conjunto de vectores de características definidos num espaço 26-dimensional. Cada vector de dimensão 26 é composto por 12 MFCCs e um termo da energia no segmento em causa, mais 13 elementos correspondentes às "primeiras derivadas pesadas", encontrando-se então definido num espaço $\Re^{26}$ por,

$$x_n = \left(x_n^1, \ldots, x_n^{13}, \Delta x_n^1, \ldots, \Delta x_n^{13}\right) \tag{2.12}$$

2.5.2 Pré-Processamento do sinal visual

Para o reconhecimento automático da fala, as características visuais de relevo são aquelas que apresentam uma maior variação. Analisando as variações na face humana, no momento em que se fala, fica-se com a convicção que as regiões de interesse para um reconhecedor são a zona da boca (lábios) e a zona acima dos olhos, as sobrancelhas. Decidiu-se ficar apenas com os lábios como região de interesse (ROI), uma vez que o orador em causa apresentava poucas variações a nível das sobrancelhas.

As técnicas para extracção das características visuais existentes podem ser divididas em três grandes grupos:

- Métodos de alto nível orientados ao modelo paramétrico[1];
- Métodos de baixo nível orientados ao *pixel*;
- Métodos híbridos.

Nos métodos de alto nível o objectivo é extrair das sequências de imagens um modelo paramétrico, estatístico, normalmente do contorno dos lábios. Técnicas como a da triangulação com os olhos e o nariz [85][86], a dos contornos activos [87][88], ou de *templates* deformáveis [89][90], são as mais usuais. O modelo gerado é aplicado à boca do falante e depois de ajustado, os seus parâmetros são utilizados como características visuais. A principal desvantagem destes métodos deve-se a que o modelo gerado poder excluir informação linguística relevante. O tempo excessivo que pode demorar a encontrar o modelo para cada imagem (*frame*) é outro dos problemas usuais que surgem nestes métodos.

Por outro lado, os métodos de baixo nível procuram definir (localizar) a região de interesse (boca do falante, etc.) através de transformações na imagem com técnicas de abordagem ao *pixel*. Essas transformações têm como objectivo potenciar eventuais diferenças entre a região de interesse e as restantes, de modo a tornar mais robusto e efectivo o processo de detecção. Nestes métodos são usados os valores resultantes das transformações do valor da luminância[2] dos *pixels* como características visuais. O método mais conhecido para pré-processamento de características visuais, com o objectivo de detectar a região de interesse, é a exclusão do vermelho [91]. Contrariamente aos métodos de alto nível, nestes a informação linguística relevante não é excluído, uma vez que, toda a região de interesse (da boca) é considerada. Este tipo de método torna-se desaconselhado para sistemas a funcionar em tempo real, uma vez que precisam de bastante poder de processamento.

Os métodos híbridos combinam as características dos métodos anteriores. De modo a obter características visuais, estes métodos utilizam parâmetros geométricos e transformações de imagem [72][92][93].

Após um estudo dos métodos existentes, e analisando o tempo disponível e a dificuldade da tarefa, decidiu-se optar por realizar uma abordagem ao *pixel* (método de baixo nível).

[1] Também conhecido por modelo geométrico
[2] Ou outra unidade equivalente

2.5.2.1 Localização da Região de Interesse (ROI)

O primeiro passo para extracção de características visuais consiste na localização da região de interesse, a boca. Este processo deve realizar-se para todos os *frames* de imagem do ficheiro vídeo.

Da análise da base de dados verificou-se que nas várias gravações as condições foram constantes, mesmo dentro da cada gravação, tornando mais vantajosa a utilização de um método orientado ao *pixel*, pelo facto de ser o que menos elimina a informação linguística. Embora esta abordagem precise de maior poder de processamento na fase de extracção de características, é mais eficaz do que as baseadas na geometria ou contornos dos lábios, na localização da ROI, embora seja mais influenciada pelas variações de iluminação (é menos robusta).

Para reduzir o tempo de processamento, a primeira operação realizada foi a de reduzir a resolução da imagem para metade, de 720×576 para 360×288.

2.5.2.1.1 Exclusão das componentes de cor da imagem

A zona labial é a zona da face que contém maior intensidade de vermelho. Uma vez que a imagem pode ser dividida nas suas três componentes principais, vermelho, verde e azul (*red*, *green* e *blue*, RGB), a abordagem inicial foi a de trabalhar sobre a componente que em principio deveria ter mais informação discriminante, a componente vermelha.

Após análise de imagens da componente vermelha, ficou a ideia que a simples utilização desta componente não seria eficaz na detecção dos lábios.

Foi então decidido testar o método de exclusão de vermelho [91]. A ideia base deste método prende-se com o facto que uma vez que a cor da face, incluindo os lábios, é predominantemente vermelha, qualquer contraste que possa ocorrer será observável nas componente verde ou azul.

Após a separação das três componentes da cor, o método efectua a análise,

$$\log\left(\frac{G}{B}\right) \leq \beta \tag{2.13}$$

Onde G é a componente verde da imagem e B a componente azul.

A escala logarítmica tem por objectivo obter maior contraste em zonas da imagem mais uniformes, ou seja, ter um maior poder discriminativo. Assim, variando o valor de β, altera-se a gama de valores a utilizar na pesquisa da zona labial. Neste caso, da exclusão do vermelho, as zonas com mais forte componente vermelha (os lábios) ficam com os valores dos *pixels* abaixo do valor máximo β. A imagem resultante da aplicação deste método revela, para o falante em questão, que o método não detecta a ROI como desejado. Por este processo, a tarefa de identificação dos lábios seria demasiado complexa e pouco eficaz.

O passo seguinte foi a realização da técnica semelhante à anterior mas para a componente azul. A técnica de exclusão do azul tinha sido utilizada com bons resultados [67]. O método excluía a componente azul através da seguinte transformação:

$$\log\left(\frac{R}{G}\right) \geq \delta \tag{2.14}$$

Como na técnica de exclusão do vermelho, a escala logarítmica visava obter um maior valor do quociente R/G. O parâmetro de pesquisa δ é obtido pela análise de algumas imagens das três sessões de gravação, estimando-se posteriormente um valor mínimo. Com esse mínimo é realizada uma pesquisa na imagem dos lábios, sendo considerados como pertencentes aos lábios qualquer *pixel* acima desse valor.

Da análise dos resultados obtidos verificou-se que a técnica não teve o sucesso desejado.

2.5.3.1.2 Detecção do eixo de Simetria

Partiu-se à procura de novas soluções face à dificuldade encontrada na localização da ROI. A face humana tem uma fisionomia essencialmente simétrica e pode ser traçado um eixo de simetria que intercepta a zona labial. De facto, durante a gravação da base de dados as condições mantiveram-se praticamente inalteradas, o falante manteve uma posição estável em relação à iluminação e ao material de aquisição, com a cabeça na posição vertical e sem oscilações significativas, a determinação desse eixo de simetria poderia facilitar a detecção posterior da zona labial. Deste modo foi seguida a metodologia de detecção [67] que apresentou bons resultados.

O algoritmo baseia-se na diferença entre blocos da imagem, procurando-se o valor mínimo, o que acontece quando a imagem é simétrica. O tamanho dos bloco é fixo e de 100×288 *pixels*.

Da direita para a esquerda, selecciona-se um primeiro bloco (Bloco1), e calcula-se a diferença *pixel* a *pixel*, do primeiro bloco com o bloco imediatamente à sua direita (Bloco2), este último invertido segundo o eixo horizontal. Os valores das diferenças e da coordenada do limite direito do primeiro bloco são guardadas numa variável, no caso de a diferença ser menor que o valor anteriormente guardado.

O processo repete-se até ao limite direito da imagem. Após cada iteração novos blocos são obtidos deslocando os anteriores dois *pixels* para a direita.

Quando o processo concluiu, a variável referida contém o valor mínimo da diferença que corresponde ao eixo de simetria encontrado e a coordenada desse eixo.

O método é extremamente rápido e eficaz quando o falante mantém uma posição frontal e estável em relação ao material de aquisição, estando a sua face sob uma iluminação uniforme e constante, que é o caso da base de dados gravada.

Nas Figura 2.11 e Figura 2.12 pode-se visualizar o fluxograma do algoritmo implementado e o resultado do mesmo, respectivamente.

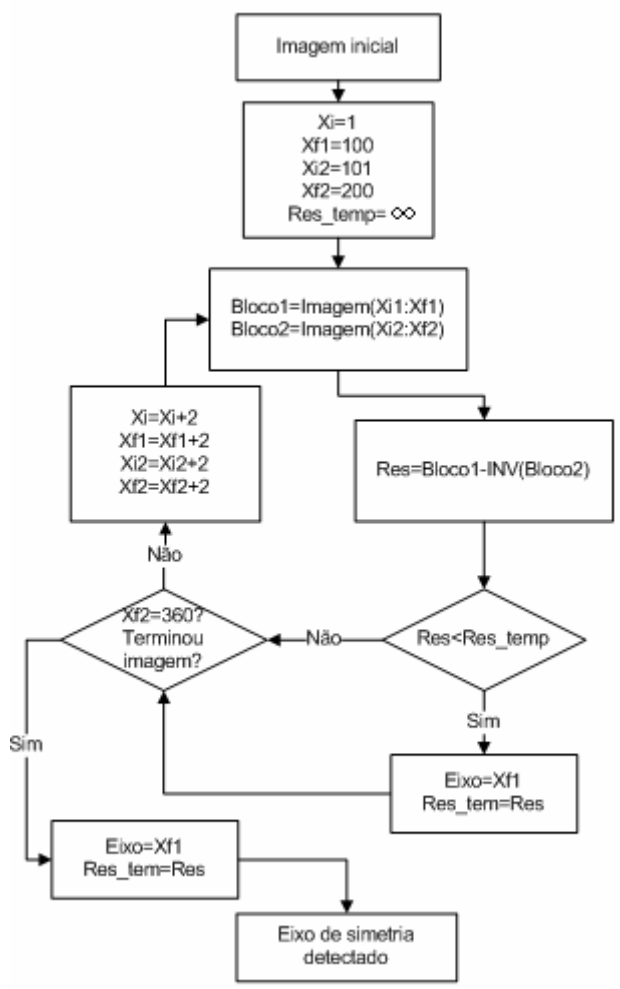


Figura 2.11 - Fluxograma do algoritmo de detecção do eixo de simetria

Figura 2.12 - Representação do eixo de simetria obtido para várias imagens da base de dados LPFAV2 a) *Frame* da sessão de 22122003 (Boca fechada); b) *Frame* da sessão de 22122003 (Boca aberta); c) *Frame* da sessão de 23122003 (Boca fechada); d) *Frame* da sessão de 23122003 (Boca aberta); e) *Frame* da sessão de 24112003 (Boca fechada); f) *Frame* da sessão de 24122003 (Boca aberta).

Da análise dos resultados obtidos pode-se concluir que o método determina com bastante rigor o eixo de simetria e dá garantias quanto à sua utilização.

2.5.3.1.3 Detecção dos pontos extremos dos lábios

O passo seguinte consiste em procurar os lábios através de um varrimento vertical ao longo do eixo de simetria uma vez que a única zona predominantemente vermelha nessa linha é a zona dos lábios.

Efectuando uma pesquisa sobre o eixo de simetria, de cima para baixo, procurámos, em cada, o primeiro *pixel* com valor acima de δ, considerado o primeiro ponto pertencente ao lábio superior, e o último *pixel* com valor acima de δ, ou seja, o último ponto localizado no lábio inferior.

Mais uma vez recorrendo à simetria da face humana, verifica-se que os pontos extremos horizontais, isto é ponto extremo esquerdo e ponto extremo direito, dos lábios se encontram, teoricamente, a meio dos pontos extremos verticais, os pontos superior e inferior dos lábios.

Deste modo, efectua-se novamente uma pesquisa em cada imagem. O varrimento é feito entre os pontos extremos verticais, partindo do eixo de simetria, procurando numa primeira passagem para a esquerda o último *pixel* com valor acima do valor mínimo δ, considerado o ponto extremo esquerdo. Depois outro varrimento é efectuado para a direita, armazenando a posição do último *pixel* com valor acima de δ, ou seja, ponto extremo direito. Obteve-se assim os pontos extremos dos lábios e definimos a região de interesse.

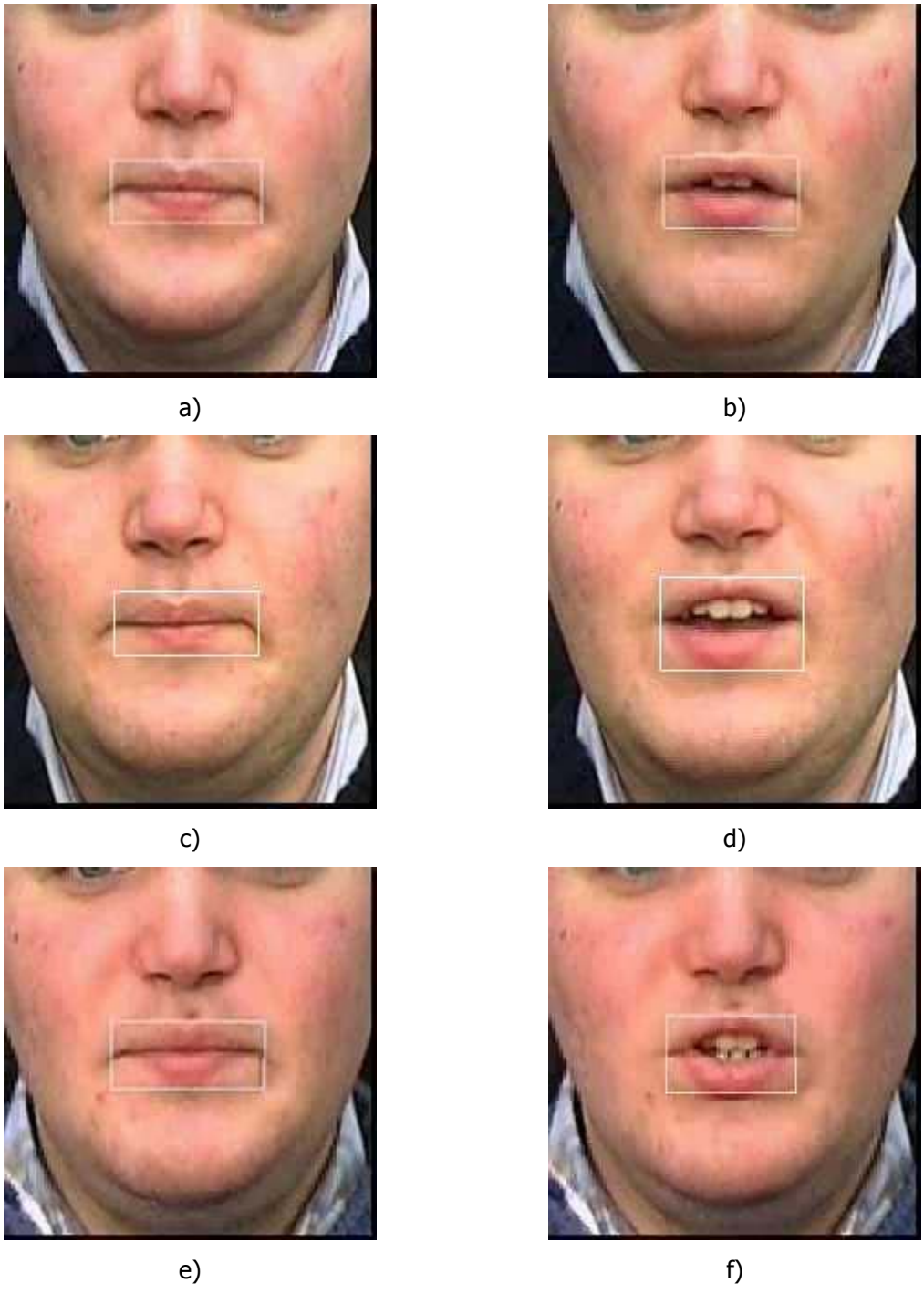


Figura 2.13 - Pontos extremos dos lábios e definição da região de interesse da base de dados LPFAV2 a) *Frame* da sessão de 22122003 (Boca fechada); b) *Frame* da sessão de 22122003 (Boca aberta); c) *Frame* da sessão de 23122003 (Boca fechada); d) *Frame* da sessão de 23122003 (Boca aberta); e) *Frame* da sessão de 24112003 (Boca fechada); f) *Frame* da sessão de 24122003 (Boca aberta).

2.5.3.2 Extracção das características visuais

Apesar de uma abordagem inicial em que se pretendia modelar os contornos dos lábios através de um modelo paramétrico baseado em *splines* [94], esta técnica foi colocada de lado devido ao facto de excluir informação relevante (como já foi referido anteriormente), relacionada com a posição da língua e dos dentes. Outro motivo para não se ter seguido este método prende-se com a necessidade da adaptação do modelo a cada *frame* de vídeo, o que iria tornar o processo moroso. Assim foi seguida a abordagem de baixo nível, orientada ao *pixel*.

Foram desenvolvidas duas versões. A primeira apenas considerava as características baseadas nos *pixels*. A segunda consiste num método híbrido, no qual são combinados as características baseadas nos *pixels* e os parâmetros retirados de um modelo de alto-nível (altura e largura).

O principal objectivo foi obter um método que conseguisse representar o movimento dos lábios com o número mínimo de parâmetros (características) de modo a reduzir o tempo de processamento (acelerar o processo de extracção de características e os processos de treino e descodificação subsequentes). O número de características adoptado foi 26, número igual ao do sinal acústico, o que facilitou os posteriores desenvolvimentos do sistema. Este valor foi assumido em função do compromisso entre velocidade de processamento e do poder discriminativo em relação aos dados, possibilitando um treino robusto dos parâmetros num processo não muito moroso.

2.5.3.2.1 Método de baixo nível (versão 1)

A primeira versão para extracção das características visuais baseou-se apenas numa abordagem ao *pixel*. A cada *frame* da ROI aplica-se a transformada discreta do co-seno (*Discrete Coussine Transform* - DCT).

A DCT gera um conjunto de coeficientes, pouco correlacionados entre si, com cardinal $M \times N$, sendo esta a dimensão da matriz a transformar. De um modo analítico, para uma matriz $M \times N$ tem-se,

$$c[i,j]=\sqrt{\frac{2}{M}\cdot\sqrt{\frac{2}{N}}}\cdot k[i,j]\cdot\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}p[m,n]\cdot\cos\left(\frac{(2m+1)\cdot i\cdot\pi}{2\cdot M}\right)\cdot\cos\left(\frac{(2n+1)\cdot j\cdot\pi}{2\cdot N}\right)$$

com,

$$k[i,j]=\begin{cases}\frac{1}{2} & se\ i=j=0\\ \frac{2}{\sqrt{2}} & se\ i\ ou\ j=0\\ 1 & se\ i\ e\ j\neq 0\end{cases} \qquad (2.15)$$

Uma vez que a janela que define a ROI varia devido ao movimento da boca, decidiu-se normalizá-la obtendo-se uma imagem de dimensões 32×32 *pixels* como a apresentada na Figura 2.14.

a) b)

Figura 2.14 - Imagem normalizada com 32x32 pixels. a) Boca aberta; b) Boca fechada

Aplicando a DCT à imagem normalizada, seguido de uma máscara adequada (Figura 2.15), obtemos uma compressão da informação, resultando o vector de características desejado de dimensão 26.

A referida máscara tem como função seleccionar os coeficientes considerados de maior relevância, representados a escuro na Figura 2.15, como correspondentes aos que melhor caracterizam a imagem.

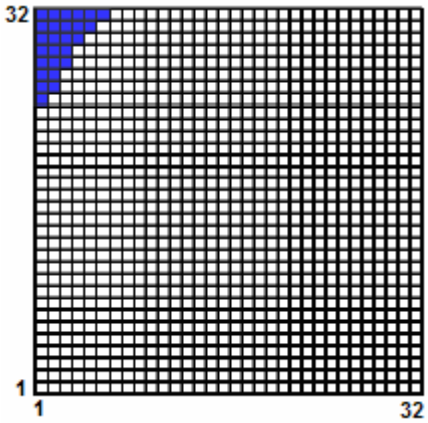


Figura 2.15 - Máscara aplicada ao resultado da DCT - versão 1

Deste modo, de cada *frame* de vídeo é retirado um vector de características visuais composto pelos 26 coeficientes que se apresentam de seguida:

- Os primeiros seis coeficientes da primeira linha da matriz dos coeficientes;
- Os primeiros cinco coeficientes da segunda linha da matriz dos coeficientes;
- Os primeiros quatro coeficientes da terceira linha da matriz dos coeficientes;
- Os primeiros três coeficientes da quarta linha da matriz dos coeficientes;
- Os primeiros três coeficientes da quinta linha da matriz dos coeficientes;
- Os primeiros dois coeficientes da sexta linha da matriz dos coeficientes;
- Os primeiros dois coeficientes da sétima linha da matriz dos coeficientes;
- O primeiro coeficiente da oitava linha da matriz dos coeficientes;

A base de dados foi adquirida a uma taxa (*frame* rate) de 25 imagens por segundo, logo, os vectores de características são extraídos a cada 40ms. Para sincronizar os vectores visuais com os vectores de características acústicas, estes extraídos a cada 10ms, foi necessário realizar uma interpolação dos vectores de características visuais. Este sincronismo é vital para facilitar o processo de treino e descodificação na abordagem *multi-stream*.

2.5.3.2.2 Método híbrido (versão 2)

Esta segunda versão foi desenvolvida com o objectivo de procurar saber qual a influência da inclusão de um modelo paramétrico, ou seja, um modelo comparativo. A nível da abordagem ao *pixel*, tal como no método anterior é utilizada a DCT, com uma máscara semelhante ao método anterior (Figura 2.16), mas neste caso apenas são seleccionados 24 coeficientes.

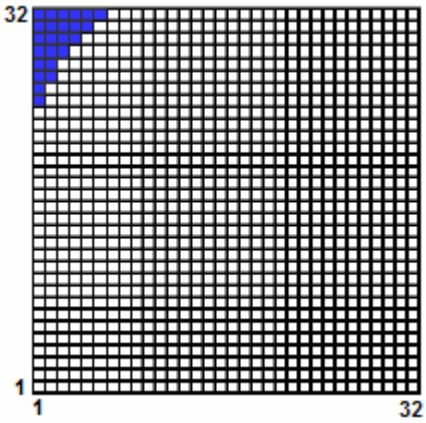


Figura 2.16 - Máscara aplicada ao resultado da DCT – versão 2

O modelo paramétrico que se decidiu utilizar foi o mais simples possível e consistiu apenas nos dois parâmetros que se seguem:

- A altura do rectângulo que define a ROI, ou seja, a distância entre o ponto fronteira superior do lábio superior e o ponto fronteira inferior do lábio inferior;
- A largura do mesmo rectângulo, ou seja, a distância entre o ponto extremo esquerdo e direito dos lábios.

Deste modo, de cada *frame* de vídeo é retirado um vector de características visuais composto pelos 26 coeficientes que se apresentam de seguida:

- Os primeiros seis coeficientes da primeira linha da matriz dos coeficientes;
- Os primeiros cinco coeficientes da segunda linha da matriz dos coeficientes;
- Os primeiros quatro coeficientes da terceira linha da matriz dos coeficientes;
- Os primeiros três coeficientes da quarta linha da matriz dos coeficientes;
- Os primeiros dois coeficientes da quinta linha da matriz dos coeficientes;
- Os primeiros dois coeficientes da sexta linha da matriz dos coeficientes;
- O primeiro coeficiente da sétima linha da matriz dos coeficientes;
- O primeiro coeficiente da oitava linha da matriz dos coeficientes;
- A largura dos lábios;
- A altura dos lábios.

2.6 Modelação Acústica e Visual

A modelação das unidades básicas de fala definidas ao nível da palavra foi possível uma vez que a aplicação de reconhecimento é realizada sobre um vocabulário relativamente pequeno (68 palavras), e o conjunto de treino apresenta suficientes repetições de cada palavra.

A modelação ao nível da palavra é vantajosa pois é mais precisa do que a modelação por fonemas. Isto acontece pois a segmentação palavra a palavra introduz de uma maneira natural uma contextualização que é importante no reconhecimento contínuo da fala, embora ainda subsista o problema da co-articulação entre as palavras

A segmentação da base de dados torna-se muito mais fácil e menos morosa do que se fosse feita ao nível do fonema.

O uso de fonemas exige a presença de léxicos o que constitui um problema uma vez que são frequentemente desadequados e pouco fiáveis. Ao se usar palavras, tal não é necessário o que constitui uma vantagem e torna o processo de descodificação mais fácil.

Estes modelos baseiam-se no formalismo estatístico dos Modelos de Markov Não Observáveis (HMM), sendo que uma outra solução possível seria o uso de Redes Neuronais (*Neural Networks* – NN) ou uma abordagem híbrida. Os HMMs são eficazes na captura da natureza temporal de processos tais como a fala. Este modelo proporciona uma forma natural e altamente fiável de reconhecer a fala num vasto conjunto de aplicações e permite que sejam introduzidas restrições gramaticais de uma maneira relativamente simples.

Existem igualmente diversos algoritmos eficientes para as tarefas de treino e reconhecimento o que é altamente positivo em situações em que necessitamos de lidar com grande volume de informação. No entanto, os HMMs apresentam uma capacidade limitada de reconhecer padrões complexos que envolvam dependências superiores à primeira ordem em sequências de informação e são altamente sensíveis às variações nas condições de gravação.

As Redes Neuronais apresentam uma boa capacidade de aprendizagem e de generalização e constituem uma boa escolha para tarefas de reconhecimento estático. Além disso, elas têm a característica de degradarem os resultados lentamente na presença de ruído. Porém, este tipo de abordagem não consegue modelar de uma forma eficaz fenómenos dinâmicos, encontrando-se as Redes Neuronais restritas a decisões locais.

Uma vez que o processo da fala contínua é um acontecimento essencialmente dinâmico, optou-se por usar os modelos de Markov não observáveis na modelação audio-visual do sistema.

2.6.1 Modelos de Markov Não Observáveis

A metodologia de reconhecimento de fala baseada nos modelos de Markov não observáveis[1] é uma das mais utilizadas. A teoria dos HMM teve origem no início do século XX (pelo matemático Markov) mas só foram desenvolvidos os algoritmos de treino em 1966 por Baum [95], e por primeira vez aplicada em reconhecimento de fala por Jelinek em 1969 [96]. Só posteriormente, a partir de 1975, com o desenvolvimento tecnológico, começaram a ser apresentadas aplicações com regularidade [97][98][78], sendo na década de 80 que foram publicados um conjunto de artigos explicitando a teoria básica [99][100][101][102], que permitiu que esta metodologia se tornasse tão popular.

Os principais motivos para o uso de HMMs na modelação da fala são as suas próprias características. As características que apresentam mais variabilidade prendem-se com a sua forma e velocidade[2]. Em relação à forma, o sinal de fala pode apresentar uma estrutura

[1] Também conhecidos por modelos escondidos de Markov
[2] Variabilidade temporal

imprevisível, sendo por isso considerado um sinal não determinístico. Deste modo, o sinal de fala é frequentemente representado por um modelo estatístico, caracterizado por um processo estocástico que permite estimar os seus parâmetros de forma precisa. Por seu lado, para lidar com a velocidade da fala é necessário adicionar informação temporal à modelação. Pode-se então concluir que os modelos de Markov não observáveis são um método adequado à modelação da fala, uma vez que são uma técnica estocástica associada a séries temporais.

A modelação por HMMs consiste em obter um modelo estatístico, para modelar as propriedades estatísticas e dinâmica das alterações que ocorrem entre os sons no sinal de fala. Esse modelo apresenta uma estrutura que permite descrever com precisão os processos estocáticos não estacionários que o sinal de fala apresenta.

2.6.1.1 Processos de Markov

Na descrição dos processos de Markov que se segue, por questões de simplicidade, assume-se que se vai aplicar a um sistema de reconhecimento de palavras isoladas, não sendo colocado o problema de identificação de limites temporais. A abordagem e notação apresentada são das mais seguidas a nível da literatura académica [5].

Considere-se um sistema que num determinado instante de tempo se encontra no estado S_i de entre N estados possíveis $S_1, S_2, \ldots, S_N$. A intervalos de tempo regulares o sistema evolui para outro estado ou eventualmente permanece no mesmo, em função de uma probabilidade de transição entre estados. Os diversos instantes de tempo serão apresentados por $t = 1,2,\ldots$ e o estado no instante t por q_t. A descrição probabilística deste processo estocástico requer o conhecimento dos estados ocupados nos instantes passados, ou seja,

$$P\left(q_t = S_j \middle| q_{t-1} = S_i, q_{t-2} = S_k, \ldots\right) \qquad 1 \le i, j, k \le N \tag{2.16}$$

Num processo de Markov de primeira ordem a descrição probabilística é condicionada apenas ao estado no instante anterior, podendo ser representado através de uma matriz de transição entre estados $A = \{a_{ij}\}$, independente do instante de tempo, em que cada elemento é definido por,

$$a_{ij} = P\left(q_t = S_j \middle| q_{t-1} = S_i\right) \qquad 1 \le i, j \le N \tag{2.17}$$

Esta matriz verifica as restrições estocásticas de definição de probabilidades, nomeadamente,

$$a_{ij} \geq 0, \qquad \sum_{j=1}^{N} a_{ij} = 1 \qquad 1 \leq i \leq N \tag{2.18}$$

O processo descrito é denominado de modelo de Markov observável, uma vez que a cada observação corresponde um estado. Este modelo é no entanto bastante restritivo e incapaz de ser utilizado em muitos problemas reais. Para tornar o modelo mais flexível, associa-se a cada estado uma função distribuição de probabilidade de observações. Assim, cada estado pode gerar uma observação, de entre um conjunto, de acordo com esta distribuição. A mesma sequência de observações pode assim ser gerada, com probabilidades diferentes, através de sequências diferentes de estados. A sequência de estados que gera uma sequência de observações não é conhecida, pelo que este modelo se denomina de não observável. Estes modelos encontram aplicações na solução de uma grande variedade de problemas.

2.6.1.2 Modelo de Markov Discreto Não Observável

Um modelo de Markov não observável é caracterizado através dos seguintes elementos:

- O número N de estados. Cada estado é denotado de $S = \{S_1, S_2, \ldots, S_N\}$ e o estado no instante t denotado por q_t, com $q_t \in \Gamma_q = \{1,2,\ldots,S\}, \quad t = 1,2,\ldots T$;
- A distribuição de probabilidades inicial para cada estado $\pi = \{\pi_i\}$,

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N \tag{2.19}$$

- A distribuição de probabilidades de transição entre estados definida pela matriz $A = \{a_{ij}\}$ em que,

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \quad 1 \leq i, j \leq N \tag{2.20}$$

- O número M de símbolos distintos observáveis por estado. Estes símbolos denotam-se de $V = \{V_1, V_2, \ldots, V_M\}$. Dado que neste caso existe um número finito de símbolos, o modelo representado denomina-se modelo discreto.

- A distribuição de probabilidades dos símbolos observáveis para cada estado S_j, $B = \{b_j(k)\}$ em que,

$$b_j(k) = P\left(v_k(t) \middle| q_t = S_j\right) \quad 1 \le i \le N, 1 \le i \le M, \qquad \forall t \tag{2.21}$$

Um modelo HMM é usualmente representado através da notação,

$$\lambda = (A, B, \pi) \tag{2.22}$$

Um modelo de Markov não observável do tipo discreto é em tudo idêntico a um modelo de Markov observável (tradicional), em que aos estados observáveis, foi adicionado um conjunto de novos estados, não observáveis, que possuem uma relação probabilística com os estados observáveis.

2.6.1.3 Classificação dos HMMs

Os HMM podem ser classificados segundo o tipo de funções de densidade de probabilidade em cada estado em discretos, contínuos e semi-contínuos.

Nos HMM discretos (DHMM) [103] a relação entre os estados observáveis e não observáveis traduz-se por uma probabilidade discreta. O seu princípio de funcionamento assenta na partição do espaço acústico num conjunto de sub-domínios disjuntos. Cada um desses sub-domínios é representado por um centróide, v_k, que não é mais do que um vector, cujo conjunto forma o designado *codebook*. Nesta abordagem, o vector de características que se pretende observar, o_t, através de um processo de quantificação vectorial que lhe atribui o índice do centroide mais próximo segundo uma determinada medida de distância (2.16), é associado a um sub-domínio.

$$k = \arg\min_k \{dist(o_t, \upsilon_k)\}. \tag{2.23}$$

Posteriormente é calculada a verosimilhança de observar o_t no estado S_j através do peso do sub-domínio k nesse estado, ou seja, $b_j(o_t) = c_{j,k}$.

Nos DHMM, um dos grandes problemas consiste no erro associado ao processo de quantificação vectorial. Este erro deve-se a atribuir-se uma verosimilhança constante a cada vector de características, independentemente da sua localização no interior de determinado sub-domínio. Embora seja possível reduzir o erro de quantificação, aumentando a dimensão do *codebook*, os custos computacionais deste procedimento limitam bastante a sua utilização devido ao aumento da complexidade do sistema. Outro dos problemas dos DHMM é que o *codebook* é determinado

no início do processamento, não sendo realizado qualquer processo de adaptação durante o treino do sistema, não havendo uma afinação dos seus parâmetros.

Por outro lado, os DHMM não precisam da utilização de funções na classificação, geralmente de Gauss, necessária nos SCHMM e CHMM [2], responsável pelo maior tempo de processamento nessas abordagens.

Nos HMM contínuos (CHMM) a função densidade de probabilidade é contínua optando-se geralmente por uma distribuição gaussiana. Esta abordagem pode ser vista como uma generalização da abordagem discreta, em que a cada estado, é atribuída uma função (habitualmente de Gauss) para o modelar, que é exclusiva desse estado, sendo a verosimilhança dada por,

$$b_j(o_t) = \sum_{m=1}^{M} c_{j,m} N_{x\,j,m}\ (o_t). \tag{2.24}$$

em que as M misturas são relativas ao estado j , $c_{j,m}$ é o peso da mistura m e $N_{x\,j,m}$ é uma distribuição gaussiana multivariável com o vector média μ e uma matriz covariância Σ.

Nos HMMs semi-contínuos (SCHMM) [104][105] as distribuições (gaussianas) são partilhadas por todos os estados. Essas distribuições partilhadas são geralmente designadas por *codebook*, e são definidas por um par (μ, σ), em que μ é o vector das médias e σ é o vector das variâncias. Segundo Storm [106] este tipo de modelos permite um número menor de parâmetros na modelação.

Contrariamente aos CHMM, os SCHMM partilham distribuições gaussianas usadas na modelação das densidades de probabilidade de observação permitindo assim, em princípio, reduzir o número de parâmetros livres que é necessário treinar, assim como a complexidade computacional, sem se verificar um decréscimo significativo do desempenho do reconhecedor [107]. No entanto, esta generalização apenas é possível se a dimensão da base de dados para treinar o sistema o permitir, tendo-se verificado desempenho superior dos CHMM quando a base de dados de treino permite obter modelos precisos e robustos [108][109].

A utilização dos SCHMM para o sistema projectado permite esperar que seja atingida uma solução robusta e com um desempenho elevado. Apesar de dadas as especificações do sistema qualquer uma das abordagens apresentadas ter a possibilidade de permitir um bom

desempenho, a escolha dos SCHMM tornou-se mais interessante uma vez que consiste numa abordagem intermédia entre os DHMM e os CHMM.

2.6.1.4 Os problemas elementares dos HMM

Da definição dos HMM e sua aplicação em reconhecimento da fala, existem três problemas que necessariamente se colocam:

1. O problema da Avaliação ou do teste do modelo:

   Dada uma sequência de T observações $O = \{o_1, o_2, \ldots, o_T\}$ e o modelo λ, determinar qual a probabilidade, $P(O|\lambda)$, da sequência ter sido gerada pelo modelo.

2. O problema da Descodificação ou determinação da sequência de estados:

   Dada uma sequência de T observações $O = \{o_1, o_2, \ldots, o_T\}$ e o modelo λ, determinar qual a sequência de estados $Q = \{q_1, q_2, \ldots, q_T\}$ mais provável no modelo que produziu as observações.

3. O problema da Aprendizagem, da estimação de parâmetros ou do treino do modelo:

   Dada uma sequência (ou conjunto de sequências) de observações $O = \{o_1, o_2, \ldots, o_T\}$, determinar de que forma se ajusta os parâmetros do modelo $\lambda = (A, B, \pi)$ de modo a maximizar a probabilidade da sequência dado o modelo, $P(O|\lambda)$.

Todos estes problemas devem ser resolvidos quando aplicados a um sistema. Existem múltiplas abordagens para resolver cada um deles, sendo aqui apresentada apenas as abordagens mais utilizadas.

No problema 1 o objectivo é avaliar o modo como o modelo HMM λ modela a sequência de observações O, $P(O|\lambda)$. Este problema também pode ser conhecido pelo problema de reconhecimento. No reconhecimento de palavras/fonemas, para uma sequência de observações, é dado como reconhecida a palavra/fonema correspondente ao modelo com maior probabilidade $P(O|\lambda_i)$, utilizando o método de classificação de máxima verosimilhança.

Para o cálculo desta probabilidade repare-se que, assumindo conhecida a sequência de estados, $Q = \{q_1, q_2, \ldots, q_T\}$ a probabilidade da sequência de observações ter sido gerada pelo modelo é dada por,

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(o_t|q_t,\lambda) = b_{q1}(o_1) \cdot b_{q2}(o_2) \ldots b_{qT}(o_T) \tag{2.25}$$

e por outro lado, a probabilidade da sequência de estados Q dado o modelo é,

$$P(Q|\lambda) = \pi_{q1} \cdot a_{q1q2} \cdot \ldots \cdot a_{qT-1qT} \tag{2.26}$$

A probabilidade conjunta da sequência de observações e da sequência de estados dado o modelo resulta do produto das duas probabilidades anteriores,

$$P(O,Q|\lambda) = P(O|Q,\lambda) \cdot P(Q|\lambda) \tag{2.27}$$

Finalmente, a probabilidade da sequência de observações dado o modelo resulta da soma, para todas as sequências de estados possíveis, desta probabilidade conjunta:

$$P(O|\lambda) = \sum_{\substack{todos\ os\ Q \\ possíveis}} P(O,Q|\lambda) \tag{2.28}$$

O cálculo de $P(O|\lambda)$ através do método directo apresentado é extremamente pesado computacionalmente, envolvendo um número $(2 \cdot T - 1) \cdot N^T$ multiplicações e $N^T - 1$ adições[1].

Para resolver esse problema foi desenvolvido um algoritmo para calcular esta probabilidade de um modo eficiente, através de um processo recursivo [95], a que se dá o nome de algoritmo progressivo-regressivo (*forward-backward procedure*).

Considerando a variável progressiva $\alpha_t(i)$ definida como a probabilidade de observação parcial da sequência $\{o_1, o_2, \ldots, o_t\}$ até ao instante t, conjuntamente com a ocorrência do estado S_i no instante t, dado o modelo,

$$\alpha_t(i) = P(o_1 \cdots o_t, q_t = S_i|\lambda) \tag{2.29}$$

[1] Mesmo para apenas três estados e 10 observações por estado, este valor é 1180979.

Esta variável pode ser calculada recursivamente através de:

1. Inicialização:

$$\alpha(i) = \pi_i \cdot b_i(o_1) \qquad 1 \le i \le N \tag{2.30}$$

2. Recursão:

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \cdot \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \qquad 1 \le t \le T,\ 1 \le j \le N \tag{2.31}$$

A probabilidade da sequência de observações é dada pela soma da variável progressiva para todos os estados S_i, que podem terminar a sequência, no instante final T,

$$P(O|\lambda) = \sum_{\substack{i=1 \\ i\ é\ estado\ final\ possível}}^{N} \alpha_T(i) \tag{2.32}$$

O cálculo de $P(O|\lambda)$ utilizando este método recursivo necessita apenas de $N \cdot (N+1) \cdot (T-1) + N$ multiplicações e $N \cdot (N+1) \cdot (T-1)$ adições[1].

Pode-se também considerar uma variável regressiva (*backward*) $\beta_t(i)$, que representa a probabilidade de ocorrência da sequência parcial de observações entre $t+1$ e T, $\{o_{t+1}, \ldots, o_T\}$, dado o modelo e dado que ocorreu o estado S_i no instante t

$$\beta_t(i) = P(o_{t+1} \cdots o_T | q_t = S_i, \lambda) \tag{2.33}$$

Esta variável pode ser calculada recursivamente através de:

1. Inicialização

$$\beta(i) = 1 \qquad 1 \le i \le N \tag{2.34}$$
$$se\ i\ é\ estado\ final\ possível$$

[1] Para $N = 3$ e $T = 10$ perfaz 165 operações, contra as 1180979 necessárias para o cálculo através do método directo.

2. Recursão

$$\beta_t(j) = \sum_{i=1}^{N} \beta_{t+1}(i) \cdot a_{ji} \cdot b_j(o_{t+1}) \qquad t = T-1 \ldots 1, \;\; 1 \le j \le N \tag{2.35}$$

e a probabilidade $P(O|\lambda)$ é dada por,

$$P(O|\lambda) = \sum_{\substack{i=1 \\ i \text{ é inicial possível}}}^{N} \beta_1(i) \cdot \pi_i \tag{2.36}$$

Repare-se que, a probabilidade $P(O|\lambda)$ em qualquer instante t, pode ser também calculada com ambas as variáveis progressiva e regressiva, através de:

$$P(O|\lambda) = \sum_{i=1}^{N} \beta_t(i) \cdot \alpha_t(i) \tag{2.37}$$

O problema 2 prende-se com a determinação da sequência de estados correspondente a uma dada sequência de observações. Tal como já referido, uma mesma sequência de observações pode ter sido gerada por diferentes sequências de estados. Assim, a determinação da sequência de estados correspondente a uma sequência de observações terá que obedecer a um determinado critério. Critérios diferentes conduzirão em geral a soluções diferentes.

Um dos critérios possíveis é escolher em cada instante t o estado com maior probabilidade. A probabilidade do estado S_i estar ocupado no instante t é dada por,

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} \tag{2.38}$$

sendo a melhor sequência de estados utilizando este critério dada por,

$$q_t = \arg\max_{1 \le i \le N} (\gamma_t(i)) \qquad 1 \le t \le T \tag{2.39}$$

Embora este método determine os estados com maior probabilidade em cada instante, pode gerar uma sequência de estados não válida, bastando para isso que a probabilidade de transição entre dois estados consecutivos seja zero.

Uma outra solução é escolher a sequência de estados que gera a sequência de observações em causa com maior probabilidade, $P(Q|O,\lambda)$, que é equivalente a maximizar $P(Q,O|\lambda)$. Esta maximização é realizada de forma eficiente pelo algoritmo de Viterbi:

1. Inicialização:

$$\delta_1(i) = \pi_i \cdot b_i(O_1) \qquad 1 \le i \le N \tag{2.40}$$

$$\psi_1(i) = 0 \qquad 1 \le i \le N \tag{2.41}$$

2. Recursão:

$$\delta_t(j) = \max_{1\le i\le N}\left(b_j(o_t)\cdot \delta_{t-1}(i)\cdot a_{ij}\right) \qquad 2 \le t \le T, \quad 1 \le j \le N \tag{2.42}$$

$$\psi_t(i) = \arg\max_{1\le i\le N}\left(\delta_{t-1}(i)\cdot a_{ij}\right) \qquad 2 \le t \le T, \quad 1 \le j \le N \tag{2.43}$$

3. Terminação:

$$P^* = \max_{1\le i\le N}\left(\delta_T(i)\right) \tag{2.44}$$

$$q_T^* = \arg\max_{1\le i\le N}\left(\psi_T(i)\right) \tag{2.45}$$

4. Escolha da melhor sequência -*path*:

$$q_t^* = \psi_{t+1}\cdot q_{t+1} \qquad t = T-1,\ldots,1 \tag{2.46}$$

No problema 3 pretende-se a determinação dos parâmetros do modelo $P(O|\lambda)$ (ajustamento dos parâmetros), de forma a maximizar a probabilidade. Este problema também é conhecido pelo problema do treino. A solução mais utilizada envolve a criação de um modelo inicial (por exemplo de um modo aleatório) e um método de reestimação iterativo, em que cada novo modelo gera a sequência de observações, com maior probabilidade que o modelo anterior. Utilizando o conceito de frequência de ocorrência, o novo modelo $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ é calculado a partir de (reestimação de Baum-Welch),

- $\overline{\pi}_i$ - número de vezes no que no instante $t=1$ se encontrava no estado S_i em função do número total de vezes que se verificou um estado para $t=1$;
- $\overline{\alpha}_{ij}$ - número de transições do estado S_i para o estado S_j em função do número total de transições a partir do estado S_i
- $\overline{\beta}_i$ - número de vezes no estado S_j que se observou v_k em função do número total de vezes que se observou o estado S_j

Estes valores são calculados a partir do modelo presente λ. Foi provado por Baum que este procedimento melhora a probabilidade de observação da sequência, ou seja,

$$P\left(O\middle|\overline{\lambda}\right)\geq P\left(O|\lambda\right) \tag{2.47}$$

A reestimação é efectuada até ser atingido um determinado critério de paragem. Definindo a variável intermédia $\xi_t(i,j)$, como a probabilidade conjunta de ocupar o estado S_i no instante t e ocupar o estado S_j no instante $t+1$ vem,

$$\xi_t(i,j)=\frac{\alpha_t(i)\cdot a_{ij}\cdot b_j(o_{t+1})\cdot\beta_{t+1}(j)}{P\left(O|\lambda\right)}=\frac{\alpha_t(i)\cdot a_{ij}\cdot b_j(o_{t+1})\cdot\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\cdot a_{ij}\cdot b_j(o_{t+1})\cdot\beta_{t+1}(j)} \tag{2.48}$$

A probabilidade de ocupar o estado S_i no instante t pode ser calculada somando $\xi_t(i,j)$ para todos os j,

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j) \tag{2.49}$$

As equações de reestimação podem então ser rescritas utilizando estas variáveis auxiliares:

$$\overline{\pi}_i=\gamma_1(i) \tag{2.50}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.51}$$

$$\overline{b}_j = \frac{\sum_{t=1 \quad para \quad o_t = v_k}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \tag{2.52}$$

2.6.1.5 Aplicação dos HMM em reconhecimento.

No reconhecimento baseado em HMM, existem modelos probabilísticos das entidades do vocabulário a reconhecer. O reconhecimento é efectuado determinando a probabilidade da entidade a reconhecer sido gerada por cada um dos modelos.

Para a construção de um reconhecedor de sinais de fala utilizando HMM, deve-se inicialmente construir um conjunto de modelos, um para cada classe de sons (fonemas, palavras, etc.) a reconhecer, através dos seguintes passos que constituem a fase de treino:

1. Definir o conjunto de classes de sons a reconhecer que corresponderá ao número de modelos a treinar;
2. Escolher uma topologia (o tipo de modelo, o número de estados e o número de observações por estado);
3. Obter, para cada classe, um conjunto com dimensão razoável de dados de treino;
4. Treinar os modelos.

Nas aplicações dos HMM para o reconhecimento de fala, não se usa normalmente modelos ergódicos (completamente ligados) mas sim modelos esquerda-direita, ou seja, modelos em que de um estado S_i só é possível transitar para o estado S_{i+1}, ou permanecer no mesmo estado, como mostra a Figura 2.17.

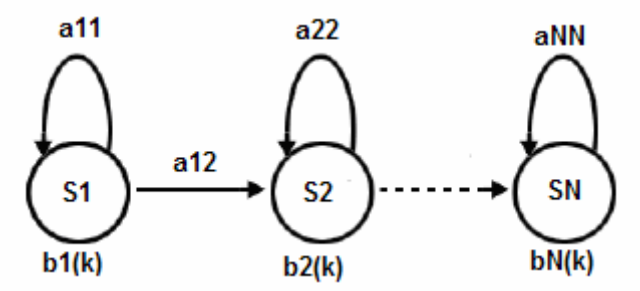


Figura 2.17 - Modelo Esquerda-Direita normalmente utilizado nas aplicações de reconhecimento de fala

Esta topologia traz algumas simplificações na estrutura dos parâmetros do modelo:

- A distribuição de estados inicial n tem apenas um valor não nulo (igual a 1), correspondente ao estado S_1;
- A matriz de distribuição das probabilidades de transição entre estados, tem, por cada linha, apenas dois valores não nulos, correspondentes a_{11} e a $a_{1(i+1)}$,

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & \ddots & a_{N-1N} \\ 0 & 0 & 0 & a_{NN} \end{bmatrix} \tag{2.53}$$

- O último estado só pode transitar para si mesmo.

2.6.1.6 HMM Compostos

Os modelos HMM compostos, são baseados nos modelos *multi-stream* HMM, e apresentam-se como uma solução muito eficaz para a integração do formalismo estatístico de um reconhecedor automático de fala audio-visual. Este tipo de abordagem permite a utilização de um modo simples do descodificador de Viterbi.

De um modo geral, a utilização de modelos *multi-stream* em sistemas multi-modais, permite combinar técnicas de modelação diferentes para os vários *streams*. O objectivo é que a combinação de diferentes tipos de modelos, conduza a um modelo final melhor que qualquer um dos modelos individuais.

Considerando dois modelos HMM individuais, o modelo *multi-stream* HMM genérico associado é apresentado na Figura 2.18. Como se pode verificar o modelo HMM resulta dos dois modelos HMM paralelos, realizando-se a recombinação dos modelos à saída. O problema destes modelos é que obriga que a segmentação dos *streams* seja coincidente (os *streams* sejam síncronos) ao nível das unidades utilizadas, o que, no caso da fala audio-visual, é muito improvável.

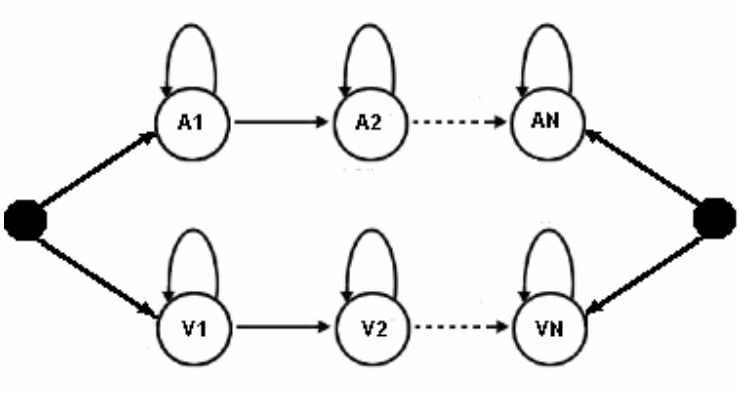


Figura 2.18 - Modelo multi-stream HMM

A estrutura do modelo *multi-stream* é bastante mais complexa, pois deve permitir todas as combinações de ocupação de estados possíveis, mesmo que os *streams* individuais não sejam síncronos. Deste modo, na utilização destes modelos o crescimento do número de *streams* implica um crescimento exponencial do número de estados.

Têm sido realizadas diversas abordagens para tornar favorável a utilização dos *multi-stream* HMM em sistemas de reconhecimento da fala com múltiplos *streams* [18][110]. Nos sistemas de reconhecimento automático de fala audio-visual a abordagem que tem obtido melhores resultados consiste numa transformação do modelo *multi-stream* genérico que resulta no modelo *product* HMM (*pHHM*) apresentado na Figura 2.19.

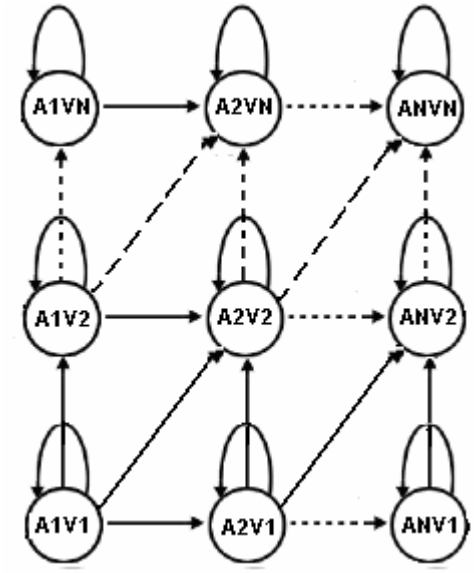


Figura 2.19 - Product HMM

Na abordagem pelos *product* HMM é permitido que modelos *single-stream* possam ser assíncronos, tendo apenas que se garantir o sincronismo nas fronteiras dos modelos. Nessas fronteiras as verosimilhanças dos modelos *single-stream* são calculadas através de uma combinação linear através da atribuição de pesos a cada *stream*.

2.6.2 Topologia dos Modelos Desenvolvidos

Como já foi introduzido na secção 2.6.1.3, após uma análise do tipo de modelos HMM existentes, discretos, contínuos e semi-contínuos, e suas vantagens e desvantagens, foi referido que se iriam implementar os modelos HMM do tipo semi-contínuo, SCHMM.

Relembrando, as principais vantagens principais dos SHCMM relativamente aos outros tipos de abordagem são:

- Permitem a obtenção de distribuições sobrepostas;
- A sua formulação permite-lhes estimar conjuntamente o *codebook* e os restantes parâmetros do modelo;
- Apresentam um número menor de parâmetros que os modelos contínuos pois utilizam um *codebook* partilhado.

É necessário definir as equações para os SCHMM implementados. A função densidade probabilidade do vector de observação em cada estado do modelo semi-contínuo, é pode ser aproximada por,

$$b_j(o_t) = \sum_{m=1}^{M_j} c_{j,m} \cdot N_{\chi_{j,m}}(o_t) \tag{2.54}$$

A distribuição gaussiana utilizada utiliza uma matriz de covariâncias diagonal. A aproximação realizada nos SCHMM deve-se ao facto de se verificar, se for bem semeado o *codebook* e os modelos, a maior parte dos estados dos modelos utilizaram uma pequena parte das gaussianas, podendo-se restringir os cálculos na determinação da verosimilhança da observação o_t nesse estado. A simplificação realizada esta representada através de $\chi_{j,m}$, e consiste num apontador, em cada estado, para as gaussianas necessárias, as quais são pesadas através dos coeficientes da mistura gaussiana respectivos, $c_{j,m}$ normalizados.

$$\sum_{m=1}^{M_j} c_{j,m} = 1 \tag{2.55}$$

As restantes expressões de reestimação são determinadas por,

$$a_{ij} = \frac{\sum_{n=1}^{N_T} \sum_{t=1}^{T^n-1} \delta(q_t = S_i, q_{t+1} = S_j)}{\sum_{n=1}^{N_T} \sum_{t=1}^{T^n-1} \delta(q_t = S_i)} \tag{2.56}$$

$$\gamma_{jm} = \frac{c_{jm} \cdot N_{\chi jm}(o_t)}{b_j(o_t)} \tag{2.57}$$

$$c_{jm} = \frac{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\delta(q_t = S_j)\gamma_{jm}(o_t)}{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\delta(q_t = S_j)} \tag{2.58}$$

$$\mu_m = \frac{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\sum_{j}\delta(q_t = S_j)\gamma_{jm}(o_t)o_t}{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\sum_{j}\delta(q_t = S_i)\gamma_{jm}(o_t)} \tag{2.59}$$

$$\sigma_m^2 = \frac{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\sum_{j}\delta(q_t = S_j)\gamma_{jm}(o_t)(o_t - \mu_m)^2}{\sum_{n=1}^{N_T}\sum_{t=1}^{T^n}\sum_{j}\delta(q_t = S_i)\gamma_{jm}(o_t)} \tag{2.60}$$

2.6.2.1 Determinação do Número de Estados

A determinação do número de estados a atribuir a cada unidade de fala é decisiva no sistema que se implementou. Ao contrário dos sistemas de reconhecimento de fala que utilizam o fonema como unidade acústica, que frequentemente assumem um valor constante de três estados para cada fonema diferente, nos sistemas cuja abordagem é à palavra, a determinação óptima do número de estados é fundamental para conseguir modelar correctamente as características fundamentais da palavra. Deste modo, cada palavra necessita de ser segmentada em N estados, dependendo das suas características. Uma vez que foi realizada uma primeira abordagem ao sinal acústico, para determinação do número de estados foram principalmente consideradas as características fonéticas[1] e linguísticas de cada palavra.

A determinação do número de estados é relevante, uma vez que, assumindo um número bastante maior do que o valor necessário, leva a um aumento de parâmetros a treinar pelo sistema, que se pode traduzir num aumento do tempo de processamento, assim como uma estimação menos robusta dos parâmetros, e um número de estados menor que os necessários

[1] O primeiro sistema a ser desenvolvido foi o sistema *single-stream* áudio.

pode tornar os modelos pouco precisos e incapazes de modelar correctamente as características principais da palavra.

A cada palavra foi atribuído um número de estados segundo a análise dos seguintes diversos factores:

- O comprimento da palavra (em número de fonemas).
- A sua duração (em segundos).
- A representação através do espectrograma.

A segunda e terceira abordagem são habitualmente substituídas por uma técnica que atribui o número de estados ao número médio de segmentos quasi-estacionários por palavra[1].

Após a análise dos resultados obtidos para os diferentes factores, foi possível chegar a um valor bastante preciso do número de estados adequado a cada palavra. Este valor é apresentado na Tabela 2.1.

Tabela 2.1 - Número de estados assumidos para cada palavra

Palavra	Nº de estados	Palavra	Nº de estados	Palavra	Nº de estados
/zero/	12	**/cinquenta/**	27	**/quadrada/**	24
/um/	6	**/sessenta/**	24	**/cúbica/**	18
/dois/	12	**/setenta/**	21	**/índice/**	18
/três/	12	**/oitenta/**	21	**/quadrado/**	21
/quarto/	17	**/noventa/**	21	**/cubo/**	12
/cinco/	14	**/cem/**	9	**/elevado/**	21
/seis/	12	**/cento/**	15	**/logaritmo/**	27
/sete/	12	**/duzentos/**	24	**/base/**	12
/oito/	12	**/trezentos/**	27	**/inverso/**	21
/nove/	12	**/quatrocentos/**	32	**/igual/**	15
/dez/	9	**/quinhentos/**	29	**/apaga/**	15
/onze/	12	**/seiscentos/**	29	**/última/**	18
/doze/	12	**/setecentos/**	29	**/tudo/**	12
/treze/	15	**/oitocentos/**	29	**/abrir/**	15
/catorze/	21	**/novecentos/**	29	**/fechar/**	18
/quinze/	18	**/mil/**	9	**/parênteses/**	30
/dezasseis/	27	**/milhão/**	18	**/vírgula/**	21
/dezassete/	27	**/mais/**	12	**/e/**	6
/dezoito/	21	**/menos/**	15	**/a/**	6
/dezanove/	24	**/vezes/**	15	**/ao/**	9
/trinta/	18	**/dividir/**	21	**/por/**	9
/quarenta/	24	**/raiz/**	12	**/de/**	9

[1] O chamado Bakis

As diferentes abordagens apresentaram valores muito aproximados, tendo sido utilizado como referência principal o método baseado no número de fonemas da palavra (são assumidos 3 estados por fonema), realizando-se depois pequenas rectificações por análise, para várias amostras de cada palavra, dos outros métodos.

2.6.2.2 *Codebooks*

Num sistema de reconhecimento de fala, cada palavra deve ser modelada de uma forma eficiente e precisa. É com base neste pressuposto que optámos pelo uso dos HMMs uma vez que proporcionam um bom compromisso entre o problema da robustez do treino e a precisão na modelação.

Nos modelos de Markov não observáveis semi-contínuos (SCHMMs), a modelação do espaço acústico ou visual é feita com base num conjunto de funções elementares, designado *codebook*, que é comum a todos os modelos.

Tipicamente, o *codebook* consiste em funções de Gauss multi-variável, as quais permitem uma modelação adequada às distribuições reais dos vectores de características.

O tamanho do *codebook* foi determinado de modo experimental e progressivo. O sistema começava com um *codebook* pequeno, e era aumentado, através de uma técnica de divisão de cada função de Gauss em duas, quando o sistema convergia.

Foi seguida a abordagem corrente que consiste em combinar linearmente as emissões gaussianas. Para tal efeito foi utilizado um conjunto de coeficientes dependente do estado do HMM.

2.6.3 Treino dos Modelos do sistema

O treino dos modelos do sistema, ou simplesmente treino do sistema consiste basicamente em determinar um conjunto de parâmetros do sistema. O treino do sistema pode ser visto como um problema de optimização desses parâmetros, seguindo um critério definido.

Um factor importante na definição da estratégia de treino é encontrar um bom compromisso entre a precisão dos modelos[1], o número de parâmetros e o tempo de processamento.

[1] Obter modelos discriminativos

A estratégia de treino seguida está dividida em duas etapas, a de semear os modelos paramétricos acústicos e visuais, e a tarefa de treinar esses modelos. Os modelos acústicos e visuais são treinados separadamente, quer na abordagem *single-stream*[1], quer na abordagem *multi-stream*[2]. Após a iniciação do sistema, vai sendo realizada a adaptação dos parâmetros juntamente com o aumento do seu número segundo a abordagem que se apresenta de seguida.

A. Iniciação
    1. Geração do *codebook* com 64 gaussianas;
    2. Iniciação dos modelos, utilizando 64 gaussianas;

B. Treino
    3. Enquanto a taxa de erro diminuir para o mesmo número de gaussianas;
        3.1 Adapta o actual conjunto de parâmetros;
    4. Duplica o número de gaussianas do sistema;
        4.1 Volta ao ponto 3;

Uma das etapas importantes é a duplicação do número de gaussianas. Sabendo que cada gaussiana é caracterizada pelo par (μ,σ), as duas novas gaussianas resultantes tem igual tem igual peso na mistura, que a que lhes deu origem. Essa duplicação consiste em manter o vector das variâncias de cada uma delas, sendo alterado o vector média do modo a deslocá-las em sentidos opostos e afastando-as pela distância de $2\cdot k\cdot\sigma$.

$$(\mu,\sigma)\rightarrow\begin{cases}(\mu-k\cdot\sigma,\sigma)\\(\mu+k\cdot\sigma,\sigma)\end{cases} \tag{2.61}$$

Deste modo, durante o treino do sistema, o *codebook* irá assumir valores de 64, 126, 512, e assim sucessivamente. Ao longo do treino os modelos vão sendo adaptados de modo a melhorar progressivamente a sua precisão.

2.6.3.1 Iniciação do Treino

Antes de se proceder ao treino do sistema, é necessário iniciar os seus parâmetros. A primeira etapa do processo consiste em semear os modelos acústicos e visuais. Porém, antes disso ser

[1] Foram desenvolvidos numa fase inicial um reconhecedor *single-stream* acústico e um reconhecedor *single-stream* visual

[2] Reconhecedor automático de fala audio-visual

feito, é necessário criar um *codebook* que contenha os valores correspondentes às médias e variâncias de cada função de Gauss (ou centróide) para cada *stream* (áudio e vídeo).

A determinação do número de centróides para cada *codebook* resulta de um compromisso entre taxa de reconhecimento e recursos disponíveis (memória, tempo de processamento).

2.6.3.2 Geração do *Codebook*

O *codebook* foi iniciado a partir do sub-conjunto[1] de dados da base de dados reservados para semear, obtendo-se 64 gaussianas.

O processo começa por realizar uma transformação do conjunto de dados seleccionado, com o objectivo de normalizar a variância global das componentes do vector de características. Seguidamente, constrói-se uma árvore de decisão binária, com uma estrutura hierárquica. O conjunto de dados é inicialmente atribuído ao nó raiz da árvore. O procedimento que se segue é recursivo e realiza-se para cada nó, contendo um conjunto não vazio de vectores de características T, até atingir o nível 6.

O algoritmo de criação do *codebook* encontra-se dividido em 6 rotinas sequenciais separadas que são realizadas de modo sequencial:

1. Cálculo das "grandes estatísticas" (média e variância) do conjunto de treino;
2. Normalização do conjunto de treino;
3. Criação de uma "árvore de decisão";
4. Distribuição do conjunto de treino (*clustering*) pela árvore de decisão;
5. Cálculo das estatísticas dos *clusters*;
6. Cálculo dos parâmetros do *codebook* (médias e variâncias de todos os *clusters*).

Todos os vectores de características do conjunto para semear são inseridos no nó raiz e encaminhado pela estrutura da árvore de acordo com a informação do centróide associado a cada nó, sendo finalmente atribuído a um dos 64 nós terminais da árvore. Para cada um desses nós, é realizada normalização, seguida de uma inversão dos vectores que lhe estão atribuídos, sendo finalmente calculados os vectores média e variância de modo a ficar definido o *coodebook* inicial.

[1] Criado a partir da segmentação da base de dados

Uma vez terminado este processo dá-se início ao passo seguinte, semear os modelos, ou seja, efectuar uma primeira distribuição das características acústicas e visuais pelos estados do modelo de cada palavra.

2.6.3.3 Método de "Semear" os Modelos

Os modelos HMM podem ser iniciados com uma quantidade reduzida de dados, no entanto, convém utilizar um número suficiente para garantir que os modelos iniciais apresentem fiabilidade para o treino. Por outro lado devem-se semear os modelos iniciais com um número aproximado de amostras por modelo diferente. Na criação do sub-conjunto utilizado para semear os modelos foi prestada atenção a esse ponto (ver secção 3.7.1).

Para o modelo do silêncio, uma vez que apresenta uma única mistura, o modelo é iniciado contando, para todos os vectores de características que lhe estão associados, de acordo com a segmentação das frases, o número de vezes que a respectiva gaussiana do *codebook* é a mais próxima do que todas as outras. Procede-se a essa contagem para todos os coeficientes, realizando-se finalmente a normalização.

As unidades de fala foram modeladas por HMMs com uma tipologia semelhante à apresentada na Figura 2.17 no caso dos sistemas *single-stream* e Figura 2.19 no caso do sistema *multi-stream*, com um número de estados por modelo conforma apresentado na Tabela 2.1. A única excepção foi o modelo para o silêncio que se modelou considerando que não apresenta nenhuma estrutura temporal, pelo que o modelo apenas apresenta um única estado, como apresentado na Figura 2.20.

Figura 2.20 - Modelo HMM para o silêncio

Para os modelos das palavras do sistema, a sucessão de vectores de características correspondendo a cada ocorrência da respectiva palavra é segmentada no número de estados correspondentes à respectiva palavra, com comprimento semelhante. Em cada estado, os coeficientes das misturas são iniciados seguindo um processo igual ao utilizado no modelo do

silêncio. No que diz respeito às probabilidades de transição, estas são iniciadas com as contagens normalizadas das transições entre estados, segundo esta segmentação linear[1].

2.6.3.4 Iteração de treino

A estratégia definida para o treino, consiste em adaptar iterativamente os parâmetros até se verificar a convergência da medida de avaliação do desempenho do sistema, neste caso, a taxa de erro (ver secção 4.1).

A metodologia implementada baseia a adaptação dos parâmetros numa aproximação, semelhante à seguida no processo de descodificação, ao critério da máxima verosimilhança. Deste modo tem-se,

$$\theta = \arg\max_{\theta} \prod_{n=1}^{N_T} p_{q\theta}\left(X^n \middle| W_c^n\right) \tag{2.62}$$

No cálculo da verosimilhança associada a cada exemplo de treino é apenas considerado o melhor caminho pertencente ao modelo correspondente a W.

Esta aproximação consiste em simplificar o treino, de modo a reduzir o tempo de processamento de cada iteração. Poderia ser realizado o treino dos modelos HMM individualmente, só que seria necessário realizar a segmentação e etiquetagem de todas as frases destinadas ao treino. Para evitar tal tarefa morosa, optou-se por uma metodologia diferente, conhecida por treino embebido. Nesta abordagem, os modelos correspondentes às palavras, que segundo a transcrição ortográfica fazem parte da frase, são treinadas em simultâneo. Deste modo, em cada iteração e para cada frase do conjunto de treino, procede-se inicialmente à composição do modelo correspondente à frase. Para além dos modelos correspondentes às palavras associadas à frase em questão, são inseridos modelos de silêncio. Estes silêncios são obrigatórios nas extremidades da frase e opcionais entre as palavras. Fica-se então com um modelos global para a frase W_c^n, composto pelos modelos individuais das palavras.

De seguida é aplicado o algoritmo de Viterbi para alinhar a sucessão de vectores de características com os estados dos HMM que integram o modelo que compõe a frase W_c^n.

[1] É realizada a repartição das *frames* de características do conjunto de treino de forma linear pela árvore de decisão, sendo atribuída à primeira *frame* o estado 1 e assim sucessivamente até ao último estado, sendo a *frame* seguinte alocada novamente no primeiro estado do modelo.

Após cada iteração com o conjunto completo de frase de treino, as variáveis que permite re-estimar os parâmetros são actualizados. Este procedimento termina assim que o valor da verosimilhança atingir um valor crítico.

2.7 Modelação Linguística

Nos sistemas de reconhecimento automático de fala após a análise lexical, é possível realizar uma análise sintáctica, onde se incorporam informações sobre a estruturação frásica e as relações entre as palavras. A análise sintáctica pode basear-se no uso de modelos de linguagem e gramáticas.

O objectivo de um sistema de reconhecimento é a procura sobre o vocabulário, de forma a encontrar o elemento que melhor representa o sinal de entrada presente. Se não for colocada qualquer restrição, a cada momento todos os elementos do vocabulário são candidatos possíveis[1], o que pode tornar o processo bastante complexo se o sistema tiver um vocabulário de grande dimensão. O objectivo da modelação linguística é, em cada momento, reduzir o vocabulário activo. O número de palavras, a cada momento, no vocabulário activo é conhecido por factor de ramificação (*branching factor*), podendo ser calculado para uma dada frase, um valor médio. Este valor pode ser utilizado para avaliar a complexidade da tarefa de reconhecimento e é conhecido por perplexidade [97][102]. Deste modo, pode-se concluir que o objectivo da modelação linguística é o de reduzir a perplexidade, o que leva a um melhor desempenho do sistema de reconhecimento que se traduz num aumento de precisão, de velocidade e da flexibilidade do vocabulário [102].

Como Jelinek afirma [97] a redução da perplexidade, no caso de um modelo de linguagem, sobre um *corpus* de teste resulta num reconhecimento com uma taxa de erros inferior, podendo-se, portanto, usar a perplexidade como uma medida da qualidade do modelo de linguagem.

As gramáticas e modelos de linguagem podem ser divididos em vários grupos:

- Gramáticas de Estados Finitos (*Finite-State Grammar*) [111]: Nestas gramáticas a lista das frases permitidas é substituída por uma representação genérica. A definição para cada instante das palavras permitidas resulta, assim, numa redução da perplexidade. Contudo, a sua forma determinística torna-as rígidas, uma vez que apenas as palavras

[1] Também designado por vocabulário activo

ou sequência de palavras definidas na gramática podem ser reconhecidas. Estas gramáticas não conseguem ordenar as palavras pela sua probabilidade de ocorrência num determinado instante, uma vez que atribuem a mesma ponderação a todas as palavras que fazem parte do vocabulário activo nesse instante;

- Gramáticas de pares de palavras (*Word Pair Grammars*) [112]: Estas gramáticas são uma variante das gramáticas de estados finitos, onde em vez de uma representação genérica se especifica as sequências de palavras permitidas através de um conjunto de pares de palavras, ou seja, para cada palavra do vocabulário indica-se quais são as palavras que lhe podem suceder.

- Modelos de linguagem estatísticos: Estes modelos examinam qual a sequência de palavras mais provável de ocorrer, em vez de especificarem a sequência de palavras permitida. Este tipo de modelação possui uma maior flexibilidade, além de permitir acomodar certas estruturas próprias da linguagem. Estes modelos são normalmente denominados de estatísticos ou estocásticos, dado que o seu funcionamento consiste na predição das sequências de palavras. A sua forma mais conhecida é o modelo *N-gram* onde se assume que a probabilidade de uma palavra depende somente das N-1 palavras anteriores. Estes modelos de linguagem estimam as sequências de palavras e as probabilidades associadas a essas sequências a partir de grandes quantidades de texto. Durante o reconhecimento, as sequências de palavras candidatas são ordenadas em função da sua probabilidade de ocorrência combinada com a probabilidade de modelarem os padrões acústicos na entrada. O vocabulário activo destes modelos pode, teoricamente, incluir todo o vocabulário, mas, na maior parte, dos casos é reduzido através do uso de limiares de probabilidade. Apesar de este modelo se mostrar efectivo em termos de reconhecimento da fala, existe na linguagem uma estrutura superior àquela que pode ser capturada por modelos *N-gram*. Se o objectivo é simplesmente traduzir o que foi dito, estes modelos funcionam bem, tornando-se no entanto insuficientes para uma análise e compreensão da mensagem que o orador está transmitindo. Por outro lado, este modelo permite qualquer sequência de palavras com alguma probabilidade. Isto permite ao modelo um nível de robustez e flexibilidade que não se consegue obter com um modelo mais rígido.

- As gramáticas baseadas em características linguísticas: Estas gramáticas baseiam-se no desenvolvimento de sistemas que compreendam o sentido das frases em vez de se restringirem a identificar a sequência de palavras produzida. Deste modo, o processo de reconhecimento da fala, orientado do ponto de vista acústico, é transformado num processo cognitivo-linguístico de entendimento e compreensão da fala. Este tipo de

gramáticas ainda se encontra numa fase embrionária, encontrando-se exclusivamente ao nível da investigação[113].

- Gramáticas independentes do contexto (*Context Free Grammars* - CFG): Estas gramáticas podem ser implementadas sozinhas ou como parte de um modelo mais amplo. São modelos determinísticos através da definição das estruturas permitidas, como no caso das gramáticas de estados finitos. No entanto, possuem capacidade de controlar e representar uma série de relações dentro da frase. Isto é conseguido através do uso de regras de reescrita e representação em árvore.

- Métodos que combinam diferentes fontes de conhecimento. Estes métodos combinam análise sintáctica (através de gramáticas independentes do contexto) com análise semântica, modelos estatísticos e outras fontes diferenciadas de conhecimento desde a aplicação, à tarefa, etc.

A implementação de sistemas de reconhecimento em aplicações com diferentes objectivos e funções permite a utilização de diferentes gramáticas e modelos de linguagem de acordo com a aplicação específica. Assim, encontramos as gramáticas de estados finitos aplicadas em realizações cuja estruturação é conhecida *à priori*, como seja, entrada de dados e controlo de equipamentos através da fala. Estas gramáticas, através da limitação do vocabulário activo, permitem aumentar a velocidade e a precisão do reconhecimento. Na maior parte dos casos, estas gramáticas são construídas pelo responsável do desenvolvimento da aplicação e adaptadas às necessidades específicas da tarefa. Como referido anteriormente, os modelos *N-gram* são utilizados na maior parte dos sistemas de reconhecimento de fala contínua. No entanto, este tipo de abordagem necessita de uma grande quantidade de texto para uma correcta estimação das suas probabilidades. Nas aplicações isso é conseguido através de uma adaptação dos modelos, previamente desenvolvidos com base num texto generalizado, ao texto produzido pelo utilizador, ou pela organização, onde o sistema de reconhecimento se encontra em uso.

Numa análise ao modelo linguístico de uma linguagem natural, este pode ser visto como resultado da combinação de quatro componentes:

- Componente simbólica.
- Componente gramatical.
- Componente semântica.
- Componente pragmática.

Os símbolos de uma linguagem definem-se como sendo as unidades mais naturais com as quais todas as mensagens são formadas, podendo ser chamados de Unidades de Mensagem Simbólicas[1] (*Symbolic Message Units* – SMU).

A gramática de uma linguagem é composta de restrições lexicais e sintácticas que descrevem a forma como as palavras são formadas de sub-palavras e como as frases derivam das palavras, respectivamente.

O objectivo da semântica é garantir que as palavras são combinadas de maneira a formarem mensagens com significado válido.

A um nível mais elevado, de abstracção, encontra-se a componente pragmática de uma linguagem que descreve como as palavras se relacionam com os oradores e o ambiente.

As restrições semânticas e pragmática são raramente usadas em sistemas de reconhecimento de fala devido à dificuldade em definir um formalismo restritivo. Por outro lado, as gramáticas são usadas em quase todos os sistemas de reconhecimento de fala contínua uma vez que as restrições lexicais e sintácticas reduzem significativamente o número de possibilidades que o módulo de descodificação precisa de considerar como hipóteses de reconhecimento.

Neste trabalho, a modelação linguística não foi focada de uma maneira extensiva. Pretendeu-se apenas com uma utilização simples, através da implementação de uma *Word Pair* e de modelos linguísticos do tipo *Bigram*, provar o benefício da sua utilização no sistema de reconhecimento de fala.

2.7.1 Restrições Gramaticais

Neste sistema, como já foi referido, a modelação linguística foi realizada através de duas abordagens, a inclusão de uma *Word Pair* e a inclusão de uma *Bigram*. A criação da *Word Pair* foi realizada com base ao conhecimento da possibilidade de a cada palavra do vocabulário, poder suceder todas as outras, desse mesmo vocabulário (ver secção 3.4.3.2).

Por seu lado, a gramática *Bigram* foi construída utilizando toda a base de dados criada, tendo sido calculadas todas as probabilidades de uma determinada palavra suceder a qualquer outra palavra do vocabulário,

[1] Representam tipicamente palavras ou sub-palavras como sílabas ou fonemas.

$$P\left(palavra_i = x \middle| palavra_{i-1} = y\right), \qquad \forall x, y \tag{2.63}$$

Essas probabilidades foram calculadas de forma automática através do software *Statistical Language Modelling toolkit da Universidade de Carnegie Mellon*, 2ª versão [114]. Na criação da *Bigram* foi utilizada toda a base de dados[1], pois, uma vez que a base de dados é de pequena dimensão, permite dar mais consistência às dependências criadas entre as palavras.

A introdução deste tipo de restrições no módulo de descodificação constitui uma mais valia no processo de reconhecimento e permite reduzir fortemente a taxa de erro.

2.8 Descodificador

O objectivo de um qualquer módulo de descodificação é o de classificar os *streams* de características que lhe são fornecidos à entrada do sistema.

No caso presente, o descodificador (também designado de reconhecedor) é alimentado com as características acústicas e visuais extraídas dos ficheiros de teste e tenta efectuar uma classificação das mesmas. Para efectuar tal classificação, baseia-se numa série de parâmetros de cada palavra previamente determinados através do módulo de treino.

O módulo de descodificação é muito condicionado pelas características de funcionamento da aplicação. Diferentes estratégias podem ser seguidas na sua implementação, dependendo da complexidade da estrutura do modelo e das restrições linguísticas da aplicação.

2.8.1 Algoritmo de Descodificação *single-stream*

O módulo de descodificação começa por inicializar os parâmetros de que necessita para efectuar a classificação. O primeiro passo consiste inicializar os modelos das palavras e o *codebook* contendo as médias e variâncias das gaussianas. Caso se pretendesse utilizar um dos modelos linguísticos gerados, eram inicializados através da introdução da *Word Pair* ou da *Bigram*.

[1] É usual utilizar apenas a parte da base de dados para treino.

Era ainda introduzido no sistema um conjunto de probabilidades auxiliares obtidas através do treino. Estas probabilidades são usadas como elementos de afinação, no cálculo das verosimilhanças, levando à obtenção de um melhor desempenho do classificador com a sua utilização.

No final desta fase a informação linguística e os modelos encontram-se embutidos na estrutura integrada do sistema. O objectivo da descodificação consiste em, dada uma sucessão de vectores de característica X_1^T, determinar a hipótese de reconhecimento correcta W_1^K, segundo um determinado critério de classificação.

O objectivo passa por determinar o caminho óptimo, que é conseguido graças a implementação de um classificador Bayesiano, deste modo,

$$W_1^K = \arg\max_W \prod_{n=1}^{N_T} p\left(X_1^T | W\right) \cdot p(W) \tag{2.64}$$

Este critério consiste no cálculo do produto da probabilidade *a priori* pela verosimilhança para todas as sucessões de palavras possíveis, identificando de seguida o valor máximo resultante o que leva à determinação da respectiva sucessão de palavras, ou seja, da frase. Se o valor de K (o número de unidades básicas) for elevado, a aplicação deste critério torna-se muito demorada uma vez que o espaço de procura cresce exponencialmente, o que torna a sua aplicação pouco viável. Para solucionar esta situação existem algoritmos de descodificação baseados no critério óptimo que através da combinação de técnicas que reduzem significativamente o espaço de procura apresentando bons resultados em sistemas de grande vocabulário.

O sistema utilizado para realizar a descodificação é o denominado descodificador de Viterbi, que realiza uma aproximação ao critério definido em (2.64), evitando que se realize o cálculo exacto da verosimilhança,

$$W_1^K = \arg\max_W \prod_{n=1}^{N_T} p_q\left(X_1^T | W\right) \cdot p(W) \tag{2.65}$$

Seja Q_W^T o conjunto de todos os caminhos de comprimento T atravessando o HMM global, com início em ω^I e final em ω^F, e atravessando por ordem indicada os modelos da palavras definidas na sucessão W. A aproximação ao valor da verosimilhança total é determinado considerando apenas a sucessão de estados q pertencente a Q_W^T, que apresentar maior verosimilhança,

$$p_q\left(X_1^T|W\right)=\max_{q\in Q_W^T} p\left(X_1^T,q|W\right) \tag{2.66}$$

O descodificador de Viterbi permite uma acentuada redução da complexidade da tarefa de descodificação, relativamente ao descodificador óptimo, sem afectar o desempenho. Por este motivo, este descodificador torna-se bastante vocacionada para sistemas utilizados em tempo real, uma vez que quando se inicia o processamento do vector de características $x_t, 1 \le t \le T$, já estão definidas as hipóteses baseadas na mesma sucessão parcial de vectores de características X_1^{t-1}.

Para implementar o descodificador de Viterbi, existem diferentes algoritmos, tendo sido implementado uma metodologia de passagem de testemunhos (*token-based Viterbi decoder*).

Cada testemunho (*token*) representa o estado do HMM global em que se encontra a hipótese de reconhecimento parcial a cada momento e a respectiva probabilidade conjunta. O processo inicia-se com um *token* no nó inicial (*token root*). Sempre que surge um novo vector de características, todos os *tokens* existentes propagam-se aos respectivos estados sucessores sendo actualizado o seu conteúdo em função da informação do HMM global e das características da observação. Em relação a cada estado apenas sobrevive um *token* que apresenta a hipótese de reconhecimento parcial com maior probabilidade, sendo os restantes eliminados antes de se proceder a nova transmissão de *tokens*. O processo repete-se e no final da descodificação, o único *token* sobrevivente encontra-se no estado final contendo informação sobre a frase descodificada.

2.8.1.1 Cálculo da Verosimilhança

Teoricamente, a verosimilhança é calculada apenas pelos valores da probabilidade estimada para a Cadeia de Markov e Densidade de Probabilidade de Observação. Tal não acontece uma vez que, na prática, chegou-se à conclusão de que são obtidos melhores resultados se forem introduzidos outros valores de afinação no seu processo de cálculo. Assim, a fórmula de cálculo da verosimilhança varia de acordo com a sua localização no algoritmo, nomeadamente quando o *token*:

- Se mantém no mesmo estado da palavra:

  A verosimilhança é calculada somando ao valor actual os logaritmos da probabilidade de transição e da densidade de probabilidade de observação para esse estado.

- Avança de estado dentro da mesma palavra:

  O cálculo é efectuado somando ao valor actual os logaritmos da probabilidade de transição do estado actual e da emissão de estado do estado seguinte.

- Salta para outra palavra:

  Ao valor actual da verosimilhança adiciona-se a soma dos logaritmos da probabilidade de transição e da emissão de estado para o primeiro estado da palavra seguinte bem como do logaritmo do coeficiente que é referido como aWW e do valor do coeficiente WP (*Word Penalty*). O primeiro parâmetro é obtido através do módulo de treino enquanto que o parâmetro WP é afinado empiricamente através da análise de resultados.

- Salta de uma palavra para *garbage model* intermédio:

  O cálculo da verosimilhança é efectuado somando ao valor actual os logaritmos da probabilidade de saída da palavra actual e da emissão do *garbage model*. A este resultado adiciona-se o valor logarítmico do parâmetro aWG determinado aquando do treino.

- Salta de uma palavra para fim:

  Neste caso, a verosimilhança restringe-se apenas ao valor actual adicionado dos logaritmos da probabilidade de saída da palavra actual e da emissão do *garbage model*.

- Salta do *garbage model* intermédio para uma palavra:

  O valor final é calculado somando ao valor já existente da verosimilhança os logaritmos do parâmetro aGW, do WP e por fim da emissão de estado da palavra.

- Mantém-se no *garbage model* intermédio:

  A verosimilhança resume-se ao valor presente ao qual é adicionado o logaritmo do parâmetro aGG e da emissão do *garbage model*.

- Salta do *garbage model* inicial para uma palavra:

  O cálculo processa-se de forma semelhante à do ponto anterior, sendo que o parâmetro aGW é, neste caso, substituído pelo coeficiente aBW calculado através do processo de treino.

- Mantém-se no *garbage model* inicial:

  A verosimilhança é calculada de forma análoga à do ponto anterior, sendo utilizado agora o parâmetro aBB e não aGG.

- Salta de um *garbage model* para o fim:

  Finalmente, no último caso, o valor da verosimilhança é obtido adicionando ao valor actual o logaritmo da emissão do *garbage model*.

2.8.1.2 Modelo de Passagem de *Tokens*

Depois de calculada a verosimilhança num qualquer ponto do algoritmo, é necessário averiguar se o seu novo valor é maior do que outro já existente para essa palavra e estado. O algoritmo implementado verifica se essa verosimilhança já foi calculada. Se a resposta for negativa então é guardada na estrutura do *token* o resultado da verosimilhança. Se já tiver sido calculado um novo valor, este vai ser comparado com o que já existia anteriormente. Caso o novo valor seja inferior o *token* não é substituído, caso contrário, um novo *token* contendo o novo valor de verosimilhança e com a história do seu *token* pai é criado.

Como foi referido, o uso e da passagem de *tokens* é facilmente implementado através do uso de HMMs. Considere-se um HMM com três estados. Assumindo que cada estado do modelo da palavra é capaz de armazenar um *token* e que contém um determinado valor. A esse *token* damos o nome de *token* pai. O *token* pai pode criar até 3 *tokens* filhos: um que se mantém no mesmo estado, outro que passaria de um estado para o seguinte, e um terceiro que retrocederia para o estado anterior. Quando os filhos são criados, o algoritmo passa a informação do pai para o *token* filho juntamente com os valores da cadeia de Markov e da densidade de probabilidade de observação relativos a cada estado. O algoritmo Viterbi encarrega-se de verificar se o novo *token* origina um caminho com maior verosimilhança ou não. Uma versão simplificada do algoritmo de descodificação é mostrada de seguida:

Inicialização:

Cada estado contém um *token* com valor inicial v=0

Algoritmo:

Para i=1 até n *Tokens* faz

Para cada estado i faz

- Tenta passar uma cópia do *token* para os estados ligados j, juntando ao seu valor v os valores da cadeia de Markov e da densidade de probabilidade de observação;
- Se o novo *token* tiver v maior do que o v pré-existente, o *token* filho substitui o anterior;

fim

fim

Terminação:

Analisa os *tokens* que estão no estado final, o *token* que apresentar o valor v mais elevado é o *token* com maior verosimilhança

2.8.2 Descodificação *multi-stream*

A descodificação num sistema *multi-stream* é análoga à apresentada para o sistema *single-stream*. Todo o processo é semelhante, verificando-se apenas diferença no modelo utilizado. Essa diferença traduz-se na substituição do modelo HMM normal pelo modelo HMM composto. Como apresentado na secção 2.6.1.5, a tipologia do modelo HMM *multi-stream* utilizado foi o *product* HMM, dadas as suas vantagens. Como foi referido anteriormente, a utilização dos modelos compostos permite a utilização do algoritmo de Viterbi de modo simples. A principal diferença do sistema *multi-stream* implementado, relativamente ao sistema *single-stream*, tem a ver com o cálculo da verosimilhança global que passa a ser calculada com base no melhor caminho.

Na sua implementação, as diferenças fundamentais do algoritmo *multi-stream*, em relação ao *single-stream* baseiam-se, por um lado no acesso aos estados que agora vai ter que obedecer à estrutura do modelo HMM composto, e, por outro lado, ao cálculo da verosimilhança da observação do sinal em cada estado, que agora vai ter que avaliar as duas funções que modelam localmente as verosimilhanças das observações correspondentes aos dois *streams*, acústico e visual.

Como será de esperar o tempo de processamento do sistema será superior ao tempo dispensado na abordagem *single-stream*, no entanto, utilizado a tipologia para o modelo

composto, em que se considera apenas o melhor caminho, perde-se um pouco de resolução, ganhando-se no entanto em rapidez de processamento. Considerando os dois *streams* do sistema implementado o tempo de processamento na abordagem *multi-stream* baseada nos *product* HMM será aproximadamente três vezes inferior à abordagem *single-stream*.

2.8.3 Parâmetros do sistema

Na implementação do algoritmo de descodificação foi considerado um conjunto de parâmetros com objectivos específicos no classificador automático de fala.

2.8.2.1 *Word Penalty*

Este parâmetro tem como função penalizar ou beneficiar o valor da verosimilhança quando se salta para a nova palavra. A utilização deste parâmetro num sistema de reconhecimento verifica-se quando se pretende alterar a relação existente entre o número de palavras inseridas e apagadas introduzidas pelo classificador.

2.8.2.2 *Beamwidth*

O *Beamwdith* (BW) é um parâmetro que tem como objectivo a limitação dos caminhos que o modelo global pode seguir ao logo de uma frase. Essa limitação traduz-se num ganho de tempo de processamento[1] pois as possibilidades de pesquisa ficam reduzidas. Este parâmetro é especialmente benéfico em sistemas de reconhecimento de grande vocabulário, onde uma pesquisa de todos os caminhos se poderia tornar num processamento impraticável.

Este parâmetro deve tomar o menor valor possível, mas sempre com um critério bem definido de modo a que não sejam eliminados caminhos correctos. Esta evidência é principalmente notada em sistemas em que os modelos não são suficientemente discriminativos.

2.8.2.1 Factor linguístico

Para melhorar o desempenho de um sistema de reconhecimento, podem ser inseridos no módulo de descodificação outros parâmetros.

[1] O BW também é utilizado no algoritmo de treino com a mesma finalidade

Um dos parâmetros que é habitualmente utilizado nos sistemas de reconhecimento automático de fala é o factor linguístico, mais conhecido por α [1].

O α pode tomar valores entre 0 e 1 e é usado no cálculo das densidades de probabilidade de observação de cada estado (emissões de estado), multiplicando-se o seu valor pelo valor da emissão do estado. O seu objectivo é "suavizar" as curvas gaussianas de modo a torná-las mais ou menos abrangentes e específicas.

Embora este parâmetro acabou por não ser explorado no sistema desenvolvido de sistema, tendo sido mantido um valor constante.

[1] Em [67] foi designado por CNOA

CAPÍTULO 3

A BASE DE DADOS LPFAV2

Este capítulo descreve o processo de definição, recolha, segmentação e estruturação da base de dados audio-visual de fala, LPFAV2, para o Português Europeu. Na sua caracterização será apresentada informação relativa ao conteúdo linguístico, informante, condições e equipamento de gravação, assim como ao pré-processamento dos sinais adquiridos.

3.1 Considerações gerais

As principais motivações na construção da base de dados de fala LPFAV2 foram, o reconhecimento robusto da fala, e uma nova solução tecnológica para algumas pessoas com deficiências físicas. De um modo geral, pode-se afirmar que o uso das características visuais juntamente com as características acústicas têm ganho uma elevada importância sendo uma das técnicas cada vez mais promissoras no âmbito do reconhecimento robusto da fala [115][116].

Um dos requisitos mais importantes no desenvolvimento de sistemas de reconhecimento de fala é uma base de dados com os materiais adequados para treino e teste. O tamanho da base de dados é decisivo para alcançar os resultados pretendidos, por isso, recolher e processar os dados suficientes para criar uma base de dados útil não é uma tarefa imediata. No caso das

bases de dados multi-modais, estas tarefas tornam-se mais difíceis devido aos vários *streams* existentes e à enorme quantidade de informação [117][118]. Por este motivo, não é surpreendente que seja escasso o número de bases de dados audio-visuais. Para o caso particular de aplicações de reconhecimento de fala para o Português Europeu, antes da criação da LPFAV2 só existia outra base de dados audio-visual conhecida pelo autor[67]. De realçar que a LPFAV2 é a única base de dados projectada especialmente para uma aplicação cujo utilizador tem uma deficiência motora específica (distrofia muscular).

Para além de suporte à investigação de reconhecedores de fala audio-visuais, esta base de dados pode também ser um valioso recurso para o estudo de aplicações específicas de suporte a pessoas com deficiências físicas. Sendo uma base de dados de um só falante é óbvia a limitação para esses estudos, as conclusões retiradas devem ser cuidadosamente analisadas no processo de generalização das mesmas.

3.2 As Bases de Dados Audio-visuais

Em contraste com a enorme quantidade de bases de dados acústicas de fala, o número de base de dados audio-visuais é muito reduzido. Um dos motivos é que esta tecnologia, audio-visual, é relativamente mais recente, mas o principal deriva dos problemas e desafios que são colocados relativamente à criação de uma base de dados audio-visual, como são a sua aquisição, armazenamento, tratamento, distribuição, etc., que não surgem de forma tão importante nas bases de dados de fala apenas com a componente áudio. Por exemplo, a aquisição dos dados visuais com qualidade, sincronizado com o sinal áudio, requer uma enorme capacidade de *hardware*[1], que muitas vezes nem com uma alta qualidade de compressão de imagem se consegue evitar.

A maior parte das bases de dados audio-visuais são resultado do esforço de pequenos grupos de investigação universitários, ou de investigadores isolados, com poucos recursos. Por este motivo, a maior parte destas bases de dados apresentam os seguintes limitações [118][120][53]:

- São de um único falante, ou com um número de falantes pequeno;
- Tem uma duração pequena;
- Geralmente são criadas para tarefas de reconhecimento simples.

[1] Capacidade de memória e de processamento

Da primeira surge logo a ideia que ao terem um número limitado de falantes é afectada a possibilidade de generalização dos métodos desenvolvidos baseados na sua utilização. A segunda leva a que muitas vezes os dados sejam insuficientes para treinar correctamente os modelos estatísticos. Da última constata-se que as aplicações para as quais estas bases de dados têm sido criadas são situações de reconhecimento de fala de palavras isoladas ou conectadas[1], ou seja, bem diferente da naturalidade pretendida com a fala contínua. Estas limitações levaram a que surgisse uma grande diferença entre a complexidade da tarefa de reconhecimento, e o estado de desenvolvimento, dos sistemas baseados apenas no áudio e dos sistemas áuvio-visuais.

3.2.1 Bases de Dados de pequeno ou médio vocabulário

A primeira base de dados audio-visual utilizada num sistema de reconhecimento de fala surgiu em 1984 e foi gravada por Petajan [51]. Apenas um falante dizia 2 a 10 repetições de uma das 100 palavras do vocabulário em inglês, incluindo dígitos e letras, em condições de iluminação controladas. Desde então diversos investigadores dedicaram parte do seu trabalho a construir novas bases de dados audio-visuais.

Foram desenvolvidas uma série de bases de dados para o desenvolvimento e estudo do reconhecimento audio-visual de consoantes, vogais ou transições entre elas. Em 1996, Adjoudani e Benoit [55] criaram uma base de dados de um só falante, para a língua Francesa, com um *corpus* de 54 palavras (*non-sense words*). Foram criadas outras com semelhantes propósitos, Su e Silsbee [54], Robert-Ribes [121], Teissier [71] todos para a língua francesa e Czap para o Húngaro.

O maior número de bases de dados audio-visuais que se podem encontrar são as de dígitos isolados ou ligados. A base de dados Tulips1 [122]foi desenvolvida para sistemas de reconhecimento de palavras isoladas, tendo sido utilizada em diversos desenvolvimentos de sistemas de reconhecimento [123][81][124][125], e contém os dígitos de "um" até "quatro" gravados por 12 falantes diferentes. A base de dados M2VTS [126], que foi desenvolvida para aplicações de verificação do falante, contém os dígitos de "zero" a "nove" e foi gravada por 37 diferentes falantes, sendo a língua Francesa a mais utilizada nessas gravações. Esta base de dados foi ainda utilizada em sistemas de reconhecimento isolado de dígitos [72][127]. Em 1999 foi terminada a extensão da M2VTS (ficou conhecida por XM2VTSDB), passando a conter 295 falantes na língua Inglesa [128]. Foram ainda desenvolvidas outras bases de dados como a NATO RSG 10 (um único falante) utilizada para o reconhecimento isolado de dígitos [129] e duas para reconhecimento de dígitos conectados. Recentemente foram gravadas duas bases de

[1] Também designadas por ligadas

dados orientadas ao reconhecimento de dígitos conectados, uma delas na Universidade de Illinois (base de dados com 100 falantes) [57][130], e outra na Universidade de Clemson, a CUAVE (base de dados de 36 falantes) [154].

As bases de dados audio-visuais para reconhecimento de letras isoladas ou conectadas foram também bastante procuradas para tarefas de reconhecimento. A primeira a surgir foi gravada em 1993 para a língua Alemã por seis falantes diferentes. O seu objectivo era ajudar ao estudo e desenvolvimento de sistemas de reconhecimento de letras conectadas e foi utilizada em diversos estudos [60][131][132], embora também tivesse sido utilizada no desenvolvimento do reconhecimento isolado de letras para um só falante [133]. Neste âmbito surgiram ainda bases de dados para o Francês [70][59][134][135][136] e para o Inglês das quais destacamos a AVLetters [69][137][138].

Para além das referidas, outras foram gravadas para desenvolvimento do reconhecimento isolado de palavras. Silsbee e Bovik [139]criaram uma base de dados de um só falante com um vocabulário de 500 palavras. Foi desenvolvida uma base de dados para um sistema de palavras de comando [140][141]. A base de dados AMP/CMU, gravada na Universidade de Carniege Mellon por 10 falantes, e com um vocabulário de 78 palavras, foi utilizada por diversos investigadores [73][62]. Há ainda a referir duas bases de dados para reconhecimento de palavras isoladas gravadas na língua Alemã [142] e Japonesa [61][143]. Dada a evolução natural dos sistemas de reconhecimento audio-visual, a complexidade das tarefas foi naturalmente crescendo, deste modo foram surgindo novas bases de dados mais complexas. Uma delas foi a base de AT&T que incluía dígitos conectados e palavras isoladas do tipo consoante-vogal-consoante.

A evolução natural dos sistemas de reconhecimento automático de fala levanta problemas de complexidade cada vez maiores. Os sistemas para reconhecimento de fala contínua são a etapa seguinte no caminho para o reconhecimento da fala natural. Deste modo, foi necessário criar bases de dados de maior complexidade, que permitissem desenvolvimento de sistemas para fala contínua. A maior parte das que existem são de pequena ou média dimensão. A TIMIT [58] consiste numa base de dados de um só falante constituída pela aquisição de 150 frases, três vezes cada. Uma outra foi gravada com 400 falantes, cujas frases consistem em comandos militares ou frases de controlo [144]. Posteriormente, esta base de dados foi aumentada, mantendo-se no entanto com um vocabulário limitado de 101 palavras [130]. Foi ainda gravada uma base de dados em Português LPFAV [67], de um único falante, com um vocabulário de 38 palavras contendo os números de "zero" a "nove", ou caracteres alfanuméricos da língua Portuguesa[1] (de "a" a "z") e a palavra "espaço".

[1] Português Europeu

3.2.2 Bases de Dados de grande vocabulário

O objectivo a atingir no reconhecimento automático de fala caminha no sentido do reconhecimento de fala espontânea para uma vocabulário de grande dimensão [145] (*Large Vocabulary Continuous Speech Recognition* - LVCSR).

Até ao momento, a única base de dados audio-visual existente e que permite reconhecimento de fala independente do falante é a IBM ViaVoice™. O seu *corpus* foi gravado por 290 falantes em fala contínua. A duração da base de dados é de aproximadamente 50 horas, contendo 24325 frases transcritas e um vocabulário de 10403 palavras. Adicionalmente foi gravada pela IBM uma outra base de dados, por 50 falantes, orientada à conexão de dígitos, de modo a estudar os benefícios da componente visual nos sistemas de reconhecimento de pequeno vocabulário, a DIGITS.

3.3 Aquisição do sinal Audio-visual da LPFAV2

A LPFAV2 [146] foi gravada no Laboratório de Processamento da Fala, Electroacústica Sinais e Instrumentação (LPF-ESI)[1]. O processo de aquisição foi cuidadosamente monitorizado prestando atenção a vários requisitos:

- Ambiente controlado, tendo atenção aos ruídos ambientais envolventes;
- Iluminação dedicada;
- Equipamento de aquisição audio-visual apropriado.

Foram realizadas três sessões de gravação, espaçadas por intervalos de vários dias, em Novembro e Dezembro de 2003. Algumas pequenas diferenças podem ser encontradas no material recolhido, entre as três sessões, que não são relevantes. De modo a recolher o sinal de fala acústico o mais limpo possível e o sinal visual sem as variações da luz natural, as sessões de gravação foram recolhidos à noite, durante o fim-de-semana.

O informante[1], aluno finalista do curso de Engenharia Informática e de Computação da FEUP, encontrava-se sentado na sua cadeira de rodas, devido à doença de que padece (distrofia

[1] http://lpf-esi.fe.up.pt/

muscular), falando a uma distância de 30 centímetros do microfone que estava com uma inclinação de 15º de modo a diminuir os ruídos de respiração.

A duração efectiva de todos os dados audio-visuais é aproximadamente 125 minutos, correspondentes a 638 frases. As gravações foram realizadas de modo contínuo, sendo os materiais de leitura organizados por *scripts* com 25 frases (ver Anexo B) cada um, apresentado num ecrã LCD de um PC portátil. Entre cada 2 gravações de 25 frases fazia-se um intervalo de alguns minutos. Alguns dos *scripts* de 25 frases foram repetidos, mas apenas foram gravados uma vez por cada sessão.

O material visual foi gravado com alta resolução (qualidade) usando a câmara[2] Canon mini-DV XM-1 3CCD. Os dados foram capturados no formato *Digital Vídeo* (DV), com frequência de 25 *frames* por segundo, com resolução 720x576 *pixels*. O sinal acústico foi adquirido em sincronismo com o sinal vídeo através de um microfone, Shure Beta 58. Este sinal foi codificado no formato PCM com 16bit à frequência de 22.05KHz, com uma relação sinal-ruído (SNR) de aproximadamente 25 dB[3]. Ambos os *streams* foram transferidos e armazenados no disco de um computador em tempo real através de um ligação *Firewire*. Os ficheiros vídeo capturados inicialmente à taxa de 3700 Kbps, foram comprimidos para MPEG-4, o que permitiu uma taxa de compressão média de 1/10 sem perda significativa da qualidade. De modo a obter uma maior qualidade vídeo, sem sombras e reflexões, foi implementada uma iluminação adequada. Foram utilizados três holofotes (Lowel, Totta e Omilight), um dos quais equipado com um reflector (este reflector enviava luz indirecta uma vez que estava situado numa posição frontal ao informante para que a luz emitida não afectasse a visão do informante relativamente aos *scripts*). Para simplificar a posterior análise da imagem foi seleccionado um fundo monocromático.

As gravações foram realizadas numa parte central de uma das salas do laboratório, que tem uma área de aproximadamente 67m^2. Na Figura 3.1 é apresentada uma representação esquemática do sistema utilizado para aquisição da base de dados.

---

[1] O orador é da zona norte do país mas fala um Português normal sem pronúncia acentuada

[2] Câmara disponibilizada pelo GAUTI (Gabinete de Apoio à Utilização das Tecnologias da Informação da FEUP)

[3] Em ambientes laboratoriais controlados, ou seja, com o ruído ambiente produzido quase exclusivamente por computadores, a SNR habitual situa-se entre os 35dB e os 40dB. Uma vez que as condições de gravação foram cuidadosamente preparadas, a diferença da SNR verificada deve-se em grande parte à doença do orador.

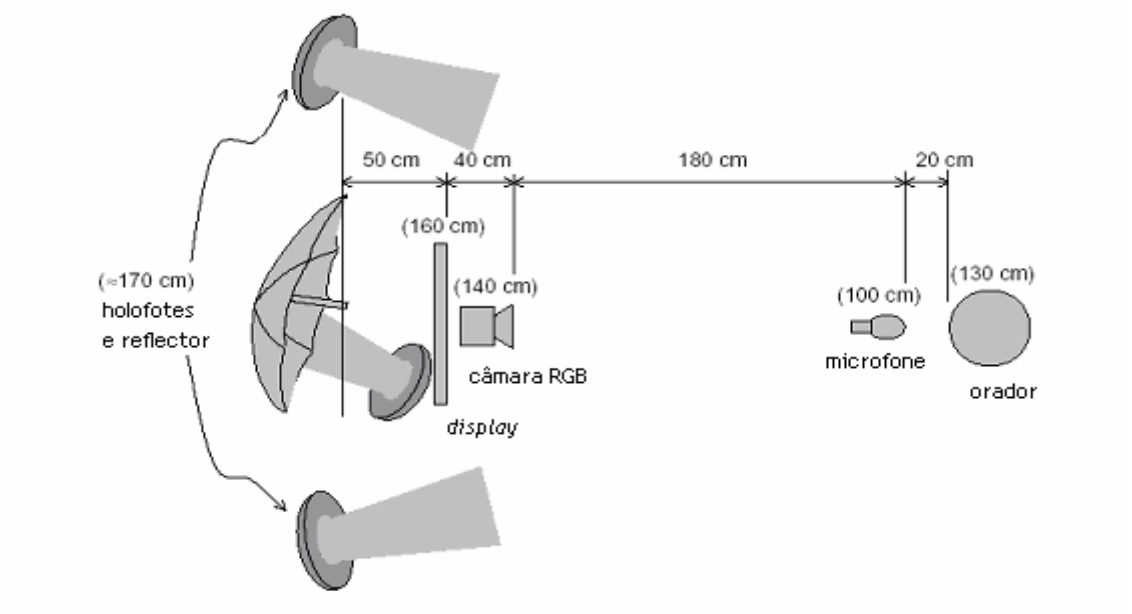


Figura 3.1 - Diagrama da instalação das disposições para aquisição da base de dados LPFAV2 (as distâncias entre parênteses consistem na distância de cada elemento ao solo)

3.4 Caracterização do *Corpus* LPFAV2

O *corpus* da LPFAV2 foi desenvolvido de modo a suportar a tarefa de reconhecimento apresentada na secção 2.2. Considerando toda a informação (dados) para treino e teste, o número total das frases gravadas é de 638 (ver Anexo A), resultando num total de 12239 palavras. A estrutura de cada frase corresponde à leitura natural de expressões matemáticas a serem realizadas por uma calculadora científica típica. As frases foram projectadas de modo a assegurar um valor mínimo de ocorrência de todas as palavras do vocabulário estabelecido, e ao mesmo tempo, um equilíbrio na ocorrência das palavras mais relevantes[1].

3.4.1 Vocabulário

O vocabulário da aplicação consiste em 68 palavras diferentes, dividas em quarto grupos: Numerais (N), utilizado para gerar os números desde zero até à escala dos milhões; Operadores Matemáticos (MO), correspondentes às operações matemáticas mais usuais de um calculadora específica; Comandos (CMM), utilizados para realizar comandos especiais; e Conectores (CNN), que consistem especialmente nas palavras de articulação para o Português. A Tabela 3.1 apresenta a lista de todo o vocabulário.

[1] São consideradas mais relevantes as palavras fundamentais para o funcionamento da aplicação (a calculadora) como são os números, os operadores matemáticos, etc.

Tabela 3.1 - O vocabulário da base de dados LPFAV2

Grupo	Palavras
Numerais	/zero/ /um/ /dois/ /três/ /quarto/ /cinco/ /seis/ /sete/ /oito/ /nove/ /dez/ /onze/ /doze/ /treze/ /catorze/ /quinze/ /dezasseis/ /dezassete/ /dezoito/ /dezanove/ /vinte/ /trinta/ /quarenta/ /cinquenta/ /sessenta/ /setenta/ /oitenta/ /noventa/ /cem/ /cento/ /duzentos/ /trezentos/ /quatrocentos/ /quinhentos/ /seiscentos/ /setecentos/ /oitocentos/ /novecentos/ /mil/ /milhão/ /milhões/
Operadores Matemáticos	/mais/ /menos/ /vezes/ /dividir/ /raiz/ /quadrada/ /cúbica/ /índice/ /quadrado/ /cubo/ /elevado/ /logaritmo/ /base/ /inverso/
Comandos	/igual/ /apaga/ /última/ /tudo/ /abrir/ /fechar/ /parênteses/
Conectores	/vírgula/ /e/ /a/ /ao/ /por/ /de/

A maior parte das palavras ocorrem aproximadamente cem vezes em todo o *corpus*. Dada a natureza da aplicação, não foi tarefa fácil conseguir este objectivo. Apenas um pequeno grupo de palavras foram mais frequentes que as restantes, ocorrendo algumas centenas de vezes. Na Figura 3.2 podemos verificar o histograma de ocorrências das palavras do vocabulário da LPFAV2.

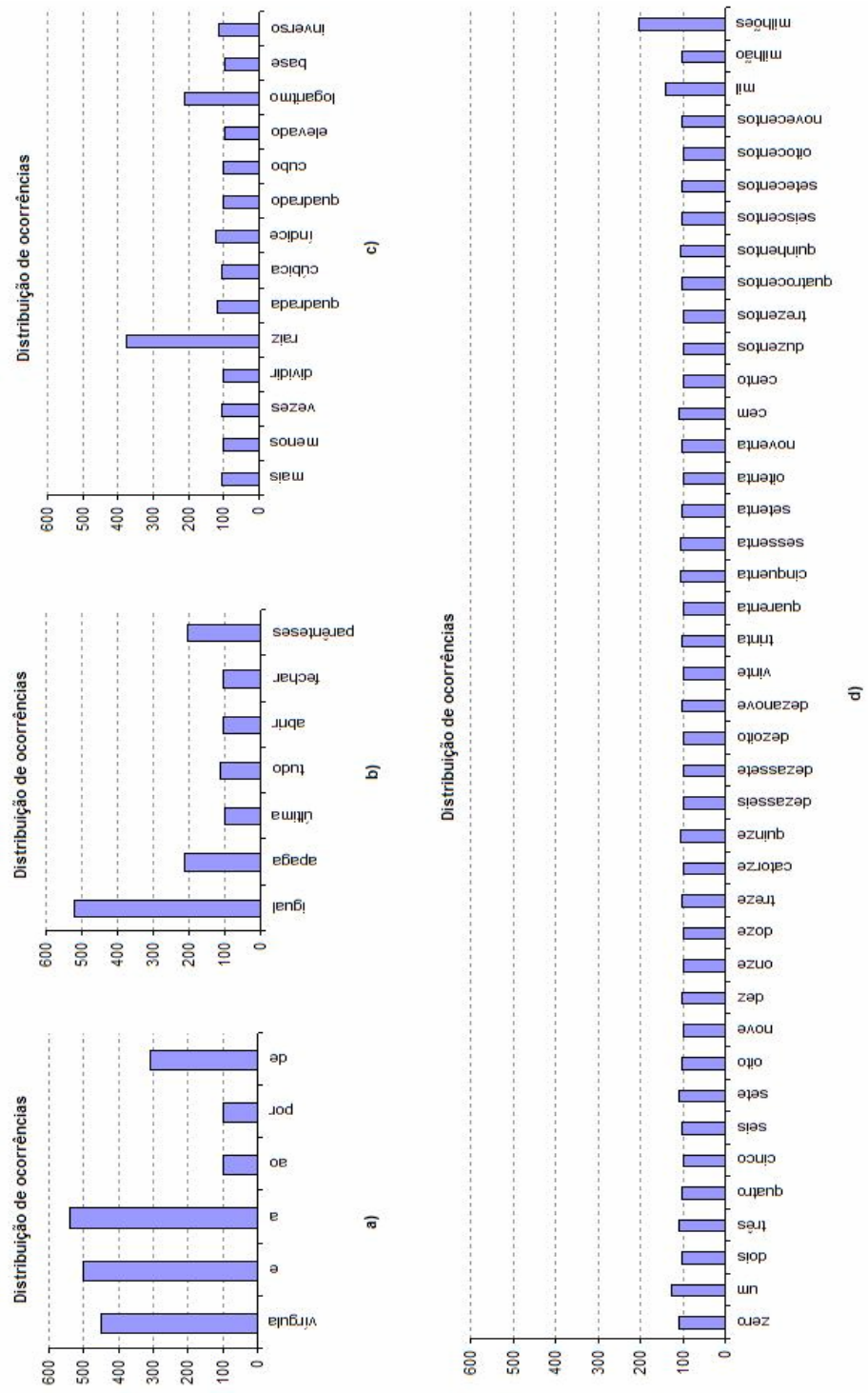


Figura 3.2 - Histograma da ocorrência das palavras apresentada na Tabela 3.1: a) Conectores; b) Comandos; c) Operadores Matemáticos; d) Numerais

3.4.2 A Gramática Generativa

A natureza da aplicação lida com frases que apresentam uma estrutura sintáctica muito rígida, seguindo um conjunto de regras restrito. A abordagem baseada no generativismo, para a elaboração de uma gramática, apresentou uma base muito sólida na base de dados adquirida para a aplicação.

3.4.2.1 O Generativismo

O Generativismo consiste numa proposta teórica de descrição linguística de regras formais e de princípios e parâmetros universais apresentada por Noam Chomsky com o objectivo de produzir todas as estruturas frásicas gramaticais de uma língua.

O programa generativista partiu do pressuposto de que a linguagem humana é um sistema de conhecimento interiorizado e procurou descrever o modo como esse conhecimento se desenvolve e se processa na mente dos falantes de uma língua. A fase da aquisição da linguagem pela criança é pois um momento que suscitou grande interesse para o programa generativista, uma vez que permite observar a evolução do desenvolvimento da linguagem. O programa inicial generativista assentava no princípio de que cada falante possuía uma gramática interiorizada composta por um dicionário mental das formas da língua e por um sistema de regras que permitiam combinar essas formas à semelhança do processamento computacional.

A principal crítica ao modelo generativista, apesar da sua permanente actualização e esforço de síntese, prende-se com o facto de se tratar de um modelo centrado na análise e descrição do sistema idealizado da língua, ou seja centrado na competência, ignorando os aspectos da performance, com tudo o que isso implica: questões de variação linguística, implicações pragmáticas, etc.

Outra das críticas tem a ver com o facto de a gramática generativista eleger como objecto de estudo a componente sintáctico-semântica da língua, não constituindo assim um modelo satisfatório de análise fonética, fonológica ou pragmática.

No entanto é de ressaltar o importante contributo do Generativismo para o estudo das questões sobre a aquisição da linguagem, enquadrado dentro de um paradigma cognitivista emergente, e o desenvolvimento de um método de investigação e de uma metalinguagem científica radicalmente inovadores.

Pelo seu papel nos estudos sobre a aquisição da linguagem e pelo desenvolvimento de um aparelho teórico e metodológico formal, rigoroso, e radicalmente diferente das propostas anteriores, o Generativismo é um modelo de análise incontornável para os estudos linguísticos em qualquer nível.

3.4.2.2 A Gramática Generativa da LPFAV2

Baseado nos modelos da gramática generativa [64], em que a estrutura de constituintes é caracterizada por um conjunto de regras de reescrita categorial (ou simplesmente regras categoriais, e na tradição europeia, também conhecidas por regras sintagmáticas), foi desenvolvido esse conjunto de regras, capaz de gerar todas as frases possíveis definidas para o sistema assim como a sua representação estrutural.

Seja G uma gramática definida através de quarto constituintes, G = (V, T, P, S) onde,

- V é um conjunto finito de símbolos denominados variáveis;
- T é um conjunto finito de símbolos denominados símbolos terminais (T é disjunto de V);
- P é um conjunto finito de regras (produções) que podem ser reescritas da forma:
    - u → w onde u e w são frases contendo qualquer combinação de variáveis e terminais e w pode ser uma frase vazia;
- S é um elemento de V denominado símbolo de partida (símbolo inicial);

A gramática G = (V, T, P, S) é definida segundo Chomsky se todas a produções forem do tipo variável→variável ou variável→terminal, logo,

- A → B C, em que A, B e C são variáveis do conjunto V
- A → "a", onde A é uma variável em V e "a" é exactamente um símbolo terminal em T.

De acordo com Chomsky, as gramáticas podem ser classificadas em:

- tipo 0 ou sem restrições: uma gramática sem qualquer tipo de restrição produção, ou seja, qualquer sequência de símbolos é possível;
- tipo 1 ou sensível ao contexto: neste caso, pode substituir-se A por B que ambos estejam no mesmo contexto e desde que B não seja vazio (ou nulo);

- tipo 2 ou sem contexto: neste caso, A pode ser substituído por B, desde não seja nulo, independentemente do contexto em que se encontrem;
- tipo 3 ou de estado finito, ou regular: neste caso, qualquer produção gramatical tem como resultado um elemento terminal do vocabulário.

A gramática generativa retirada das frases gravadas é apresentada na Figura 3.3. Esta apresenta uma estrutura de acordo com Chomsky do tipo 2, em árvore, encontrando-se nos ramos terminais palavras de uma das quatro classes (N, MO, CMM e CNN[1]) apresentadas na Tabela 3.1. As classes definidas entre parênteses são opcionais na reescrita das regras e as delimitadas por chavetas são mutuamente exclusivas.

$$Frase \rightarrow \left\{ \begin{matrix} (OCOM)\ \ FOMO\ \ MO\ \ FOMO\ \ (OCOM\ \ MO\ \ FOMO) \\ OMO\ \ (MO\ \ (OCOM)\ \ FOMO\ \ (MO\ \ FOMO\ \ OCOM)) \\ FOMO \end{matrix} \right\} OCOM$$

$$OCOM \rightarrow CMM \left\{ \begin{matrix} CMM \\ CNN \end{matrix} \right\}$$

$$FOMO \rightarrow \left\{ \begin{matrix} OMO \\ F \end{matrix} \right\}$$

$$OMO \rightarrow \left\{ \begin{matrix} MO\ \ \left\{ \begin{matrix} MO\ \ (N)\ \ (CNN) \\ CNN \end{matrix} \right\} \\ (CNN)\ \ MO\ \ (CNN) \end{matrix} \right\}$$

$$F \rightarrow ONUM\ \ \left(\left\{ \begin{matrix} CNN\ \ \left\{ \begin{matrix} N \\ MO \end{matrix} \right\} \\ MO\ \ CNN\ \ N \end{matrix} \right\} \right)$$

$$ONUM \rightarrow ON\ \ (N)\ \ (ON)\ \ (N)\ \ (ON)\ \ (CNN)\ \ (ON)\ \ (N)\ \ (ON)$$

$$ON \rightarrow N\ \ (CNN)\ \ (N)\ \ (CNN)\ \ (N)$$

Figura 3.3 - Regras da Gramática retirado do *corpus* da LPFAV2

Da análise das frases, verificamos que estas se organizam em termos de uma estrutura de constituintes. Foi então gerado o conjunto de regras capaz de criar essa estrutura e definir a sua natureza.

[1] Numerais, Operadores Matemáticos, Comandos e Conectores, respectivamente

3.4.3 Modelos Linguísticos Estocásticos

A Modelação linguística não é mais do que a arte de determinar a probabilidade de ocorrer uma sequência de palavras [147][114]. Este conceito é muito útil e utilizado em várias áreas, incluindo o reconhecimento de fala, reconhecimento óptico de caracteres, reconhecimentos de manuscritos e máquinas de tradução, entre outras.

Independentemente da unidade de fala, as restrições linguísticas estão associadas à forma como essas unidades podem ser concatenadas, em que ordem, contexto e qual o seu significado [148]. Estas regras são limitadoras do grau de liberdade de expressão do utilizador num dado sistema de reconhecimento de fala. Existe pois a necessidade de criar um equilíbrio entre a liberdade de expressão a fornecer aos utilizadores de um sistema de reconhecimento de fala e a eficiência dos modelos linguísticos através da diminuição de caminhos alternativos durante a descodificação de uma mensagem.

Uma das principais medidas do grau de restrição à liberdade de expressão pelo modelo linguístico é a perplexidade [148][151], que representa, o número médio de ramos num dado ponto de decisão em que o reconhecimento pode ser visto como a busca de caminhos num grafo de frases possíveis.

Nesta secção são apresentados os resultados de uma série de testes ao corpus da base de dados LPFAV2, com o auxílio do software *Statistical Language Modelling t*oolkit da Universidade de Carnegie Mellon, 2ª versão, de modo a avaliar a dificuldade da tarefa projectada para o sistema implementado.

3.4.3.1 As Técnicas de Modelação Linguística

No reconhecimento de fala procuram-se formalismos relativamente simples para o modelo linguístico, pois este deve ser integrado no algoritmo de descodificação.

O principal objectivo de um modelo linguístico é determinar a probabilidade de uma sequência de palavras, $\omega_1 \ldots \omega_N$, $P(\omega_1 \ldots \omega_N)$. Esta probabilidade é usualmente dividida pelas suas várias componentes, resultando,

$$\mathrm{P}(\omega_1 \ldots \omega_i) = \mathrm{P}(\omega_1) \times \mathrm{P}(\omega_2 | \omega_1) \times \ldots \times \mathrm{P}(\omega_i | \omega_1 \ldots \omega_{i-1}) \tag{3.1}$$

Se i for demasiado grande, determinar $\mathrm{P}(\omega_i|\omega_1 \ldots \omega_{i-1})$ torna-se uma tarefa complexa, logo é usual assumir que a probabilidade de uma palavra depende apenas das $N-1$ palavras imediatamente anteriores a ω_i, resultando,

$$P(\omega_i|\omega_1 \ldots \omega_{i-1}) \approx P(\omega_i|\omega_{i-N+1} \ldots \omega_{i-1}) \tag{3.2}$$

Um dos modelos que apresenta maior eficiência na integração com o reconhecimento automático de fala toma o nome de *N-grams*. Estes modelos calculam os valores aproximados das probabilidades condicionadas apresentadas em (3.1) com recurso a (3.2).

3.4.3.2 Modelos linguísticos *N-gram*

Os modelos *N-gram* são constantemente utilizados em sistemas de reconhecimento de fala pois simplificam o processo de descodificação. Com estes modelos é assumido que a probabilidade de uma palavra depende apenas das últimas N-1 palavras imediatamente anteriores, isto é, $\mathrm{P}(\omega_i|\omega_{i-N+1} \ldots \omega_{i-1})$.

Embora os exemplos mais utilizados de *N-grams* são os *Bigrams* (N=2) e os *Trigrams* (N=3) [147], este tipo de abordagem permite a formulação de modelos mais simples, sem gramática, *Word Pair* (também conhecido por par de palavras) e as unigramas:

- Sem gramática: Este modelo é utilizado quando não é possível ou não é necessário utilizar um modelo linguístico;
- *Word Pair*: Consiste na atribuição de valores binários, 0 ou 1, às probabilidades condicionais com N=2. Sempre que numa determinada língua existe a possibilidade de um par de palavras ocorrer, atribui-se 1, caso contrário atribui-se o valor zero;
- Unigramas: Neste modelo N toma o valor 1. Consiste apenas na frequência relativa de ocorrência de cada palavra numa dada língua. Este modelo é geralmente menos eficiente que o *Word Pair*.

N-grams com valores de N superiores a 3 embora com pouca frequência, também têm sido utilizados na área do reconhecimento de fala [147].

O aumento do valor de N leva a um incremento bastante acentuado da complexidade do modelo. Este aumento leva à necessidade de um número de dados muito grande para treinar correctamente os modelos. Quanto menor o valor de N, menor será a profundidade contextual

dos modelos, no entanto menor será a complexidade dos mesmos. Em bases de dados pequenas, como é o caso da LPFAV2, um valor pequeno de N é favorável e (inevitável), por este motivo nos cálculos realizados assumiu-se o valor 2, para N, ou seja, os modelos utilizados são *Bigrams*.

Os *Bigrams* ou *Trigrams* funcionam geralmente bem, mas quando o *corpus* é pequeno pode surgir um problema, muitas sequências de duas ou três palavras podem não acontecer. Para lidar com este problema existem as técnicas de *Smoothing*[149]. Estas técnicas, por exemplo, podem utilizar as probabilidades de sequências de palavras conhecidas para gerar probabilidades de sequências de palavras que não apareçam no *corpus*.

3.4.3.3 Técnicas de *Smoothing*

Como foi referido na secção anterior, na modelação linguística por *N-grams*, a probabilidade da frase $P(f)$ é expressa como o produto das probabilidades das palavras que compõe a frase e com a probabilidade condicional de cada palavra dependente das $N-1$ palavras imediatamente precedentes, por exemplo, se $f = \omega_1 \ldots \omega_n$ temos,

$$P(f) = \prod_{i=1}^{n} P\left(\omega_i \middle| \omega_1^{i-1}\right) \approx \prod_{i=1}^{n} P\left(\omega_i \middle| \omega_{i-N+1}^{i-1}\right) \tag{3.3}$$

Onde ω_i^j representa as palavras $\omega_i \ldots \omega_j$. Por exemplo, no caso das Bigrams (N=2), para estimar as probabilidades $P(\omega_i|\omega_{i-1})$ da equação (3.3) deve-se utilizar o *corpus* em texto, mais concretamente o conjunto utilizado para treino. Desse modo tem-se,

$$P_{ML}(\omega_i|\omega_{i-1}) = \frac{P(\omega_{i-1}\omega_i)}{P(\omega_{i-1})} = \frac{c(\omega_{i-1}\omega_i)/N_f}{c(\omega_{i-1})/N_f} = \frac{c(\omega_{i-1}\omega_i)}{c(\omega_{i-1})} \tag{3.4}$$

Onde $c(x)$ representa o número de vezes que a sequência x ocorre no referido *corpus*, e, N_f representa o número total de palavras. Este é o método denominado *Maximum Likelihood* (ML). Da definição rapidamente surge uma pergunta óbvia: "E se o *corpus* de treino for pequeno e não ocorrerem todos os conjuntos de palavras possíveis?" Esta observação é bastante importante e deve ser levada em conta quando se está a criar o modelo linguístico para o *corpus* de um sistema de reconhecimento de fala. Sendo o *corpus* pequeno, poderá não conter muitas dos conjuntos de palavras associados aos modelos que se pretendem desenvolver,

levando a que se assuma que a probabilidade associada é zero. Por exemplo, no *corpus* da LPFAV2, se considerarmos o par de palavras[1] "sete mil", e esse par não ocorrer no *corpus* de treino, a probabilidade associada será zero, ou seja, $P_{ML}(mil \setminus sete) = 0$. Assumir que a probabilidade da *Bigram* é zero pode levar a erros no funcionamento do sistema de reconhecimento uma vez que a probabilidade desse *Bigram* ocorrer é realmente não nula e desse modo o sistema iria ser induzido em erro. Para evitar o problema referido, existem técnicas específicas conhecidas por *Smoothing*. Estas técnicas baseiam-se no ajuste das probabilidades calculadas pelo método *Maximum Likelihood*.

Um modo simples de realizar o *Smoothing* é assumir com que cada Bigram ocorra pelo menos mais uma vez do que realmente ocorre [151], deste modo,

$$P_{+1}(\omega_i \setminus \omega_{i-1}) = \frac{c(\omega_{i-1}\omega_i)+1}{c(\omega_{i-1})+|V|} \tag{3.5}$$

Onde V é o vocabulário do *corpus* em questão. Deste modo evita-se que existam as probabilidades zero, no entanto, a probabilidade associada a esse conjunto de palavras é tão reduzida que poderá continuar a levar a erros, mas pelo menos a probabilidade do sistema de reconhecimento poderá considerar esse conjunto de palavras. Muitas outras técnicas de *Smoothing* surgiram tal como *delected-interpolate*, *linear*, *additive*, *Katz*, *Witten-Bell*, *Church-Gale*, *Jelinek-Mercer*, etc., entre outras.

3.4.3.4 Avaliação dos modelos linguísticos

Para se saber a importância de um modelo linguístico, este deve ser avaliado no *corpus* disponível. Existem diversas medidas objectivas da qualidade de um modelo linguístico ou da complexidade relativa ao *corpus* disponível. A medida mais usual é a perplexidade. Um modelo linguístico que atribui igual probabilidade a 10 palavras tem perplexidade 10. A perplexidade de um modelo linguístico pode ser definido como a média geométrica do inverso das probabilidades das palavras no *corpus* em questão [151],

$$\sqrt[N]{\prod_{i-1}^{N} \frac{1}{P(\omega_i \setminus \omega_1 \ldots \omega_{i-1})}} \tag{3.6}$$

[1] *Bigram*

A perplexidade não é mais do que uma medida aproximada do factor de ramificação médio ("*average branching factor*"). A perplexidade tem várias propriedades o que a torna muito interessante como medida de modelos linguísticos.

Uma medida alternativa, mas equivalente à perplexidade, é a entropia. A entropia não é mais que o $\log_2$ do valor de perplexidade correspondente. Considerando R uma fonte de informação que gera na sua saída uma sequência de k palavras $p^k = \{p_1, p_2, \ldots p_k\}$, então a entropia associada à fonte R, $H(R)$ representa a quantidade de informação média por palavra e é apresentada em (3.7).

$$H(R) = -\sum_{p^k} \frac{1}{k} P(p^k) \log P(p^k) \tag{3.7}$$

A entropia indica-nos o número médio de bits por palavra necessários para codificar o *corpus* utilizando um código óptimo.

3.4.3.5 A Modelação Linguística na LPFAV2

Com o auxílio do software *Statistical Language Modelling toolkit* da Universidade de *Carnegie Mellon*, 2ª versão foram realizados alguns cálculos de modo a analisar a complexidade da tarefa. Foi utilizado todo o *corpus* tendo sido obtido um valor da perplexidade, no caso de se considerarem apenas as unigramas, de 15,51 (que equivale a uma entropia de 3,96). Foram ainda geradas duas Bigrams, considerando duas técnicas de *Discounting* diferentes: *Linear* e *Witten Bell* [151][152]. O número de pares de palavras diferentes encontrados no *corpus* foi 769. A perplexidade estimada usando a técnica *Linear* e *Witten Bell* foi de 6,44 (2,69 de entropia) e 6,90 (2,79 de entropia) respectivamente. Considerando que as frases têm uma estrutura muito rígida, usando modelos linguísticos muito simples como os apresentados, levam a perplexidades muito baixas. A eficiência destas técnicas de *Smothing* pode ser confirmadas comparando as perplexidades com outras tomadas usando técnicas de *jacknife* [153].

3.5. Segmentação e Etiquetagem das frases

O *corpus* de fala tem que ter associada informação linguística acerca do seu conteúdo, de modo a poder ser utilizado no desenvolvimento de sistemas de reconhecimento de fala ou noutros tipos de sistemas de engenharia da linguagem (síntese, codificação, etc.).

As frases gravadas foram todas processadas, segmentadas e etiquetadas (marcações temporais de início e fim de palavra por observação directa dos sinais associados), manualmente ao nível da palavra, com o auxílio dos programas, Adobe Premier 6.5, Cool Edit Pro 1.2 e Praat. 4.1 Este procedimento bastante moroso foi realizado pelo autor deste documento cuja experiência na área tinha sido adquirida anteriormente na segmentação e etiquetagem de uma base de dados criada para um projecto de interface da voz para comércio electrónico.

O procedimento padrão para determinação dos limites de cada palavra seguiu-se em dois passos: primeiro foi inspeccionada a ROI através da sequência de vídeo para estabelecer as fronteiras iniciais, e depois, foi inspeccionado o sinal acústico correspondente para um ajuste fino dos limites finais. Recorreu-se ao espectrograma para dissipar algumas situações de análise mais complexa. Embora a segmentação fosse executada em um nível relativamente elevado, uma série de critérios de confiança foi estabelecido a fim de assegurar a consistência da mesma em casos de maior dificuldade na análise. Tal como esperado, algumas fronteiras não estão bem definidas surgindo palavras sobrepostas (sem fronteiras de início e fim claramente identificáveis) devido a fenómenos de co-articulação que surgem na fala contínua. As palavras mais afectadas por este problemas são as palavras de menor duração ("e", "a", "de", "ao", "por") e cuja ocorrência depende directamente de outras palavras, motivo pelo qual foram classificadas e agrupadas no sub-grupo de palavras de conexão (CNN).

Para cada frase, o resultado da segmentação e etiquetagem é registado num ficheiro de texto com um número de linhas igual ao número de palavras da frase. Na Figura 3.4 é apresentado o formato de cada linha desse ficheiro de texto.

símbolo [ss:cc , rr:bb]

Figura 3. 4 - Nomenclatura da segmentação e etiquetagem de uma palavra

Os parâmetros assumem os seguintes valores:

- símbolo: transcrição ortográfica da palavra;
- ss e cc: segundos e centésimas iniciais;
- rr e bb: segundos e centésimas finais.

A Figura 3.5 mostra o conteúdo de um ficheiro de texto da base de dados utilizado para treinar os modelos. Neste para além do cabeçalho apresenta-se a frase correspondente e a respectiva

segmentação ao nível da palavra, tal como referido anteriormente. O símbolo % foi colocado para separar os campos do registo. Nas restantes frases da base de dados, subconjuntos de teste e afinação estes ficheiros de texto são idênticos, apenas não apresentam a informação temporal de início e fim de palavra.

```
---------------
-Base de Dados-
---------------
Frase:

dois mais sete ao quadrado igual a
%
      dois      [00:48,01:10]
      mais      [01:14,01:62]
      sete      [01:78,02:33]
        ao      [02:37,02:64]
quadrado      [02:68,03:45]
     igual      [03:52,03:91]
         a      [03:92,04:08]
```

Figura 3.5 -Exemplo de um ficheiro de texto pertencente à directoria TXT FILES

O ficheiro apresentado na Figura 3.5 foi utilizado para semear os modelos.

3.6. Exame de Qualidade

De modo a obter uma melhor caracterização da qualidade acústica da base de dados gravada foram realizados uma série de testes, nomeadamente:

- Relação Sinal-ruído (SNR – *Signal Noise Ratio*) do canal áudio;
- *Clliping Rate* do sinal áudio;
- *Offset* do sinal áudio.

3.6.1. Relação Sinal-Ruído

A relação sinal-ruído foi calculada para os 638 ficheiros da base de dados. Cada ficheiro áudio da base de dados foi dividido em janelas de 10 ms contínuas, sem sobreposição. De seguida foi

calculada a energia de cada janela. Previamente foi removido o valor médio (*offset*) de todo o sinal. Foi considerado que 5% das janelas contendo o menor valor de energia apresentavam apenas ruído de linha.

A relação SNR foi então calculada, dividindo o valor médio da energia do sinal pelo valor do ruído. Este valor é posteriormente convertido para decibéis (dB) resultando na distribuição apresentada na Figura 3.6.

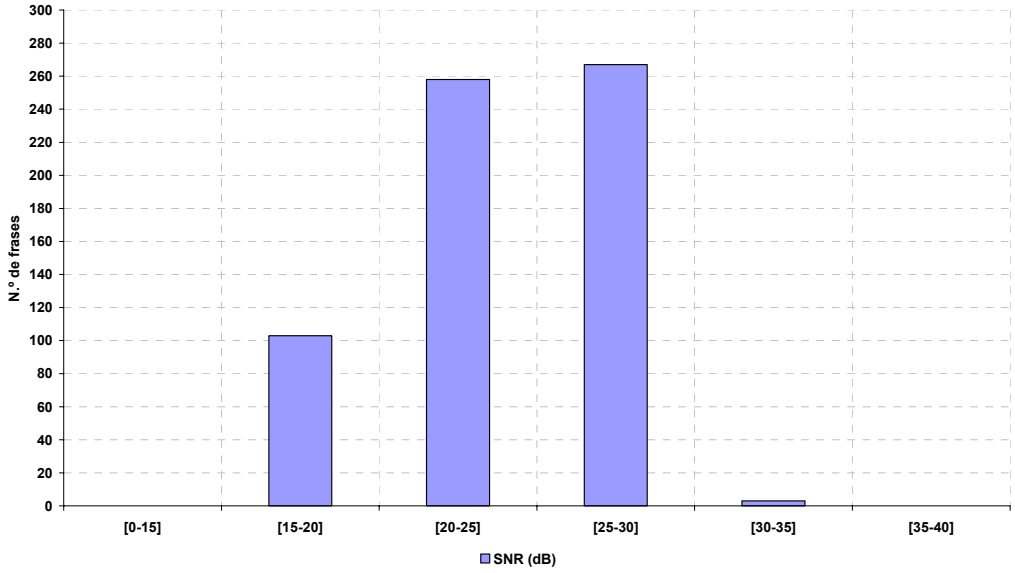


Figura 3.6 - Distribuição da relação SNR na base de dados LPFAV2

Posteriormente, na fase experimental do sistema de reconhecimento desenvolvido foi adicionado ruído aditivo branco gaussiano (AWGN – *Adictive White Gaussian Noise*) de modo a obter diversas relações SNR.

3.6.2. *Offset*

Foi determinado o valor médio de cada sinal (frase) resultando na distribuição apresentada na Figura 3.7.

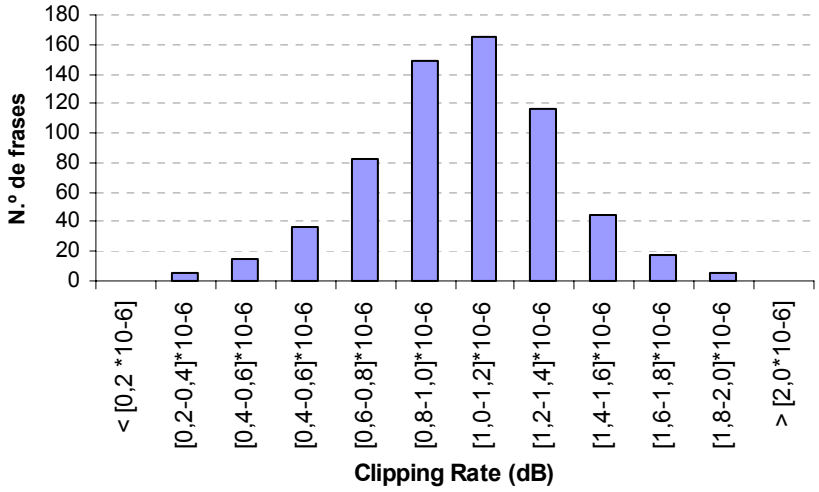


Figura 3.7 - Distribuição dos valores de *Offset*

É possível observar que o *offset* varia entre os 10^{-5} e os 10^{-6}, o que representa um baixo valor.

3.6.3. *Clipping Rate*

A *clipping rate* foi calculada para saber quantas vezes as amostras foram limitadas devido a apresentarem um valor superior ao permitido pelo formato. Para tal foi contado o número de vezes que em cada ficheiro se atingia o máximo, fazendo-se depois uma razão desse valor com o número total de amostras. Na Figura 3.8 é apresentado o histograma relativo a esta taxa.

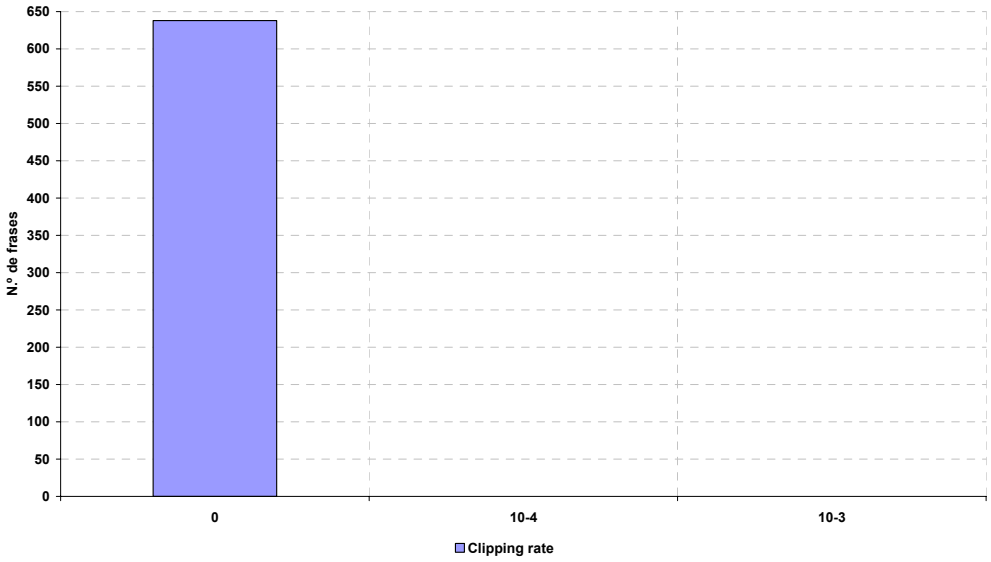


Figura 3.8 - Histograma dos valores de *Clipping rate*

O resultado deste histograma é o esperado, não havendo qualquer amostra que atinja o valor de saturação máximo permitido, uma vez que, como referido anteriormente, o falante sofre de uma doença que faz com que a intensidade do sinal de fala seja baixa.

3.7 O Pacote LPFAV2

Depois de adquirido e pré-processado o material audio-visual foi estruturado e guardado num pacote contendo os resultados da modelação acústica e visual, material para modelar a língua, documentação variada acerca das características da base de dados e outras informações auxiliares.

A base de dados encontra-se gravada em seis CDs (1 DVD), dividida em dois subconjuntos que diferem apenas no conteúdo visual. Os ficheiros com os sinais acústicos e ficheiros de texto (segmentações e etiquetagens) são exactamente iguais nos dois conjuntos. Num dos subconjuntos, com três CDs[1], os ficheiros de vídeo (.AVI) contêm as sequências de *frames* originais, contendo toda a face do falante, do modo como foram captadas e comprimidas para MPEG4. No outro subconjunto estes ficheiros vídeo contêm apenas a chamada *Region of Interest* (ROI), que consiste na área rectangular contendo os lábios. Esta operação foi realizada uma vez que a informação relevante fora desse rectângulo é comparativamente muito menor. É óbvio que esta tarefa foi realizada para permitir utilizar a base de dados de um modo mais rápido. Esta segmentação da imagem foi realizada a toda a base de dados, incluindo aos ficheiros reservados para desenvolvimento e teste.

Para cada uma das duas partes referidas, dois CDs contêm os materiais para treino e o outro contém os materiais de desenvolvimento e teste. Cada CD tem quatro directorias:

- AVI FILES;
- WAV FILES;
- TXT FILES;
- DOC FILES.

A directoria AVI FILES contém os ficheiros AVI, acima referidos, cada um relativo à frase respectiva. O nome de cada frase depende da data da gravação, do número do *script* e do número da frase dentro do *script*; por exemplo, 24112003_02_10.AVI foi gravada a 24 de Novembro de 2003, através da leitura da décima frase do segundo *script*.

Na directoria WAV FILES encontram-se os ficheiros áudio correspondentes a cada frase, que foram obtidos através da extracção dos ficheiros AVI correspondentes, com o auxílio do software AVI2WAV.

[1] Ou espaço equivalente em DVD

A directoria TXT contém os ficheiros de texto correspondentes à segmentação e transcrição ortográfica (apenas nas 200 frases seleccionadas para semear o *codebook*) de cada palavra em cada frase gravada.

A directoria DOC Files contém variada informação acerca da base de dados tal como:

- READ.ME, ficheiro que contém informação detalhada sobre a base de dados;
- LISTA_CD.TXT, ficheiro que contém o nome de todos os ficheiros vídeo do pacote;
- Outros ficheiros contendo a informação necessária ao desenvolvimento dos diversos modelos linguísticos usados no reconhecedor.

3.7.1 Definição dos Conjuntos de Treino e Teste

Com a base de dados gravada, segmentada e etiquetada, procedeu-se à separação das frases em conjuntos de treino e teste de modo a manter semelhante, em ambos os conjuntos, a distribuição das palavras do vocabulário.

Dessa forma, foi definido um conjunto de treino com 401 frases, e um conjunto de teste de 237 frases.

O conjunto de teste final representa 20% do *corpus*. Uma vez que este conjunto é já por si reduzido, não era aconselhável usar parte dele, como conjunto de desenvolvimento para efeitos de validação. Desta forma foi criado um conjunto denominado de desenvolvimento ou afinação, de modo a realizar o desenvolvimento do sistema de reconhecimento. Mantendo o mesmo princípio da criação do subconjunto de teste, este subconjunto foi criado com 20% do *corpus*.

Para o desenvolvimento do sistema de reconhecimento (ver capítulo 2), é necessário "semear" os modelos iniciais das palavras. Para tal decidiu-se que seria necessário no mínimo vinte amostras de cada palavra. Deste modo, foi necessário seleccionar um conjunto de frases do conjunto de treino formando o subconjunto semear. Este subconjunto utilizado na fase inicial de desenvolvimento do sistema, continua a ser parte integrante do conjunto de treino, sendo utilizada nas fases seguintes de desenvolvimento (treino dos modelos) do sistema. O subconjunto representa 30% do conjunto de treino.

Após esta divisão ficou-se com 60% de frases para treino e 40% para teste, como se pode verificar na Figura 3.9. Como é possível observar, o conjunto para semear os modelos é parte integrante do conjunto de treino, uma vez que após ser utilizado para semear os modelos, é

também utilizado para os treinar. Por sua vez, o conjunto de afinação pode ser considerado como parte integrante do conjunto de teste, na medida em que serve para avaliar o desempenho do sistema durante o seu desenvolvimento, não entrando no teste final do sistema.

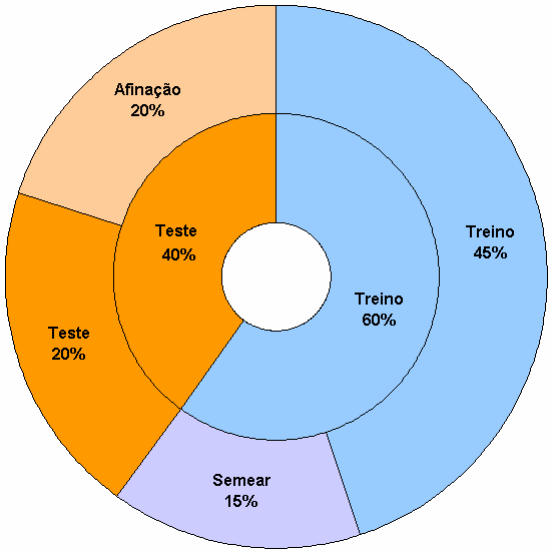


Figura 3.9 - Definição dos Conjuntos de Treino e Teste do *corpus* LPFAV2

No Anexo B podemos verificar as listas de frases escolhidas para cada um dos grupos referidos.

É importante realçar que em qualquer dos conjuntos nem todos os ficheiros foram utilizados ou então algumas das frases foram movidas para outros conjuntos no decorrer do desenvolvimento do sistema. Durante o desenvolvimento do sistema os resultados intermédios do descodificador foram sendo analisados, prestando principal relevância às frases com mais erros. Essas frases foram analisadas ao pormenor para identificar se os erros se deviam a problemas no material audio-visual, ou apenas ao funcionamento do sistema. As frases em que foram identificados problemas nos conteúdos acústico ou visual foram removidas ou trocadas de grupo, conforme o tipo do problema identificado. Os motivos mais comuns são o facto de terem sido detectados problemas associados à inteligibilidade (e.g. palavras cortadas, palavras mal pronunciadas, ruídos esporádicos demasiado intensos, etc.) ou à existência de palavras não pertencentes ao vocabulário. Estas últimas surgiram em situações em que o falante, para além de falar o texto pretendido, introduziu palavras de ligação ou, por engano na leitura do *script*, substituiu palavras da frase por outras que não pertenciam ao vocabulário definido. Estes exemplos são esclarecedores dos problemas que surgem, não só nos processos de recolha de fala (essenciais para o treino do sistema), mas também na utilização de um sistema de reconhecimento.

3.8. Considerações finais

Neste capítulo foi apresentada a base de dados multi-modal LPFAV2 criada para o desenvolvimento de sistemas de reconhecimento audio-visual de fala. Esta base de dados estabelece uma aplicação dependente do falante, suportando reconhecimento de fala contínua para o Português europeu. A base de dados tem a particularidade de ter sido gravada por um falante com distrofia muscular.

Com vista ao desenvolvimento multi-modal da aplicação foram gravados o sinal acústico e o sinal vídeo com a face do falante. Estes sinais foram tratados e incluídos num pacote juntamente com a transcrição ortográfica e etiquetagem temporal (das frases seleccionadas para semear os modelos) e ainda outros ficheiros necessários ao desenvolvimento dos modelos linguísticos.

Considerando as suas características, a LPFAV2, que combina o robustecimento do reconhecimento de fala e as tecnologias de apoio a pessoas com deficiências, pode ser uma contribuição valiosa para a investigação nestas áreas.

CAPÍTULO 4

Resultados e Evolução do Sistema

Este capítulo apresenta o desempenho dos diversos módulos do sistema desenvolvido, assim como todas as considerações relevantes assumidas durante o desenvolvimento.

Na parte inicial do capítulo é realizada uma breve abordagem aos métodos de avaliação da taxa de erro que foram assumidos para avaliar o sistema desenvolvido. São também descritas neste capítulo as diversas experiências realizadas com os dois grupos de teste.

4.1 Métodos de avaliação da taxa de erro

A avaliação correcta do desempenho de um sistema é uma tarefa fundamental para um desenvolvimento mais rápido e apurado. A avaliação dos resultados obtidos com reconhecedores automáticos de fala baseia-se na determinação de alguns valores numéricos que permitem verificar dois aspectos diferentes, a eficácia do reconhecimento e a comparação entre reconhecedores. No primeiro aspecto avaliam-se essencialmente aplicações do reconhecimento por vezes já em testes de campo. Procura-se melhorar detalhes da interacção com o utilizador ou obter especificações ou certificações da aplicação. O segundo aspecto é particularmente importante nas áreas de investigação e desenvolvimento onde são ensaiados novos algoritmos, usando as mesmas bases de dados.

4.1.1 Cálculo da Taxa de Erro (WER)

Uma vez que o sistema foi implementado considerando a palavra com unidade elementar, as variáveis utilizadas para avaliar o desempenho do sistema apresentam o resultado em função dessa mesma unidade, a palavra. A excepção é a taxa final de frases completamente acertadas.

Foi desenvolvido um programa[1], de acordo com as especificações de avaliação *standard*, que identifica e classifica os erros cometidos pelo sistema, permitindo determinar a taxa de erro – *Word error rate* (*WER*).

O princípio de funcionamento do programa consiste na procura do caminho óptimo através do qual, o resultado obtido mais se aproxima do caminho correcto.

Considerando a frase reconhecida de comprimento l_1, $\omega_1,\ldots,\omega_{l_1}$, e a frase original de comprimento l_2, $\omega_1,\ldots,\omega_{l_2}$, sendo ω_i a palavra i da respectiva frase, o algoritmo começa por analisar o comprimento das 2 frases a comparar, a original e a reconhecida. Com os comprimentos das duas frases é construída a matriz de comparação através do algoritmo que se segue:

1. Definição da matriz

$$A = \left[a_{ij}\right] \quad i = 1,\ldots,l_1, \quad j = 1,\ldots,l_2 \tag{4.1}$$

2. Inicialização da matriz

$$a_{ij} = \begin{cases} 0 & i = j = 1 \; ou \; \left(i = l_1 + 1, j = l_2 + 1\right) \\ 1 & outros \; valores \; de \; i, j \end{cases} \tag{4.2}$$

3. Comparação das frases

$$Se \left(\omega_i = \omega_j\right) \Rightarrow a_{ij} = 0 \tag{4.3}$$

A matriz de comparação representa as "distâncias" entre as palavras de duas frases, através da atribuição de distância binária "1" entre as palavras das duas frases.

Considere-se, por exemplo, que o sistema reconheceu a frase "zero vírgula cento e treze a dividir por doze vírgula vinte apaga tudo", quando deveria ter reconhecido a frase "zero vírgula

[1] Em Matlab

cento e treze a dividir por doze mais quinze apaga tudo". Na Figura 4.1 é apresentado o modo que a matriz de comparação ficaria preenchido.

		zero	*vírgula*	*cento*	*e*	*treze*	*a*	*dividir*	*por*	*doze*	*mais*	*quinze*	*apaga*	*tudo*	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
zero	1	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1
vírgula	1	1	**0**	1	1	1	1	1	1	1	1	1	1	1	1
cento	1	1	1	**0**	1	1	1	1	1	1	1	1	1	1	1
e	1	1	1	1	**0**	1	1	1	1	1	1	1	1	1	1
treze	1	1	1	1	1	**0**	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	**0**	1	1	1	1	1	1	1	1
dividir	1	1	1	1	1	1	1	**0**	1	1	1	1	1	1	1
por	1	1	1	1	1	1	1	1	**0**	1	1	1	1	1	1
doze	1	1	1	1	1	1	1	1	1	**0**	1	1	1	1	1
vírgula	1	1	**0**	1	1	1	1	1	1	1	1	1	1	1	1
*v*int*e*	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
apaga	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1	1
tudo	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**

Figura 4.1 - Exemplo para a matriz de comparação

Após ter sido preenchido a tabela, é realizada a adição de todos os caminhos possíveis desde a primeira até à última palavra. O caminho que obtiver a menor valor é o escolhido. A adição consiste em somar ao índice da palavra anterior mais uma unidade, que corresponde ao valor de um salto, obtendo-se o valor do novo índice. Na Figura 4.2 é apresentada a evolução da determinação do caminho óptimo para o exemplo apresentado.

		zero	*vírgula*	*cento*	*e*	*treze*	*a*	*dividir*	*por*	*doze*	*mais*	*quinze*	*apaga tudo*		
	0	2	4												
zero	2	**1**	3												
vírgula	4	3	**2**	×											
cento	6		4	×											
e				×	×										
treze					×	×									
a					×	×	×								
dividir						×	×	×							
por							×	×	×						
doze								×	×	×					
vírgula									×	×	×				
*v*int*e*										×	×	×			
apaga											×	×	×		
tudo												×	×	×	
													×	×	×

Figura 4.2 - Evolução do cálculo do caminho óptimo

Para determinação do WER é considerado o número total de palavras erradas, que consiste nas palavras substituídas, inseridas e apagadas. Deste modo, a taxa de erro é dada por,

$$WER = \frac{total\ \ palavras\ \ erradas}{total\ \ palavras} \tag{4.4}$$

$$WER = \frac{total\ pal.\ \ substituíd as + total\ pal.\ inseridas + total\ pal.\ apagadas}{total\ \ palavras} \tag{4.5}$$

De (4.5), observamos que o valor da taxa de erro está dependente de todos os tipos de erro inseridos pelo sistema, ou seja, erros por palavras substituídas, inseridas e apagadas. Pode-se então calcular três medidas de avaliação que são importantes para o desenvolvimento do sistema, uma vez que permitem quantificar os diferentes tipos de erro que o sistema está a cometer, permitindo avaliar melhor o seu desempenho através da variação dos seus parâmetros. Um desses parâmetros é o *Word Penalty* (WP), que permite alterar o desempenho do sistema a nível das palavras inseridas e apagadas, permitindo que esse valor seja semelhante.

Uma das medidas de avaliação é a designada taxa de substituição,

$$S = \frac{total\ \ palavras\ substituídas}{total\ \ palavras\ \ faladas} \tag{4.6}$$

Outra medida é a taxa de palavras que se perdem durante o reconhecimento, também designada por taxa de supressão,

$$D = \frac{total\ \ palavras\ apagadas}{total\ \ palavras\ \ faladas} \tag{4.7}$$

Finalmente, temos a taxa de palavras que o sistema inseriu quando não o deveria fazer. Este tipo de erros pode ser resultado de ruídos e é habitualmente conhecido por "falso alarme[1]"[156]. Esta taxa é designada por taxa de inserção,

$$I = \frac{total\ \ palavras\ inseridas}{total\ \ palavras\ \ faladas} \tag{4.8}$$

Uma outra medida utilizada para avaliar o desempenho do sistema de reconhecimento foi a taxa de frases acertadas,

$$SA = \frac{total\ \ frases\ completamente\ acertadas}{total\ \ frases\ \ faladas\ testadas} \tag{4.9}$$

4.1.2 Matriz Confusão

As medidas de avaliação do desempenho apresentadas indicam qual o desempenho global do sistema de reconhecimenho, preocupando-se apenas com determinar uma relação entre as palavras reconhecidas à saída do classificador, e as palavras correctas, discriminando apenas o tipo de erros ocorridos em relação à estrutura da frase (substituição, inserção ou apagamento).

Esses métodos de avaliação não permitem todavia avaliar os erros do sistema a um nível fundamental, que é a nível da unidade de fala do sistema, ou seja, a nível do seu vocabulário.

[1] Este termo é habitualmente utilizado em sistema de reconhecimento do falante (orador)

Para esta avaliação, existe a matriz confusão[1]. Estas matrizes consistem em comparar as frases reconhecidas com as originais, sendo registados os erros cometidos pelo módulo de classificação.

A matriz confusão permite classificar o tipo de erros cometidos pelo sistema a nível da unidade de fala utilizado, permitindo determinar se os erros cometidos são gerais, ou se estão concentrados apenas em alguns modelos. Esta avaliação é bastante importante para o sistema pois permite retirar conclusões importantes, e orientar a afinação dos parâmetros e modelos do sistema. Por exemplo, se o tipo de erros se concentra em palavras grandes ou pequenas, ou se consistem em palavras distintas com determinadas características iguais, etc. Depois de identificados esse tipo de erros pode-se determinar um conjunto de estratégias para abordar os problemas e decidir qual a melhor abordagem.

Em relação ao sistema desenvolvido, um sistema audio-visual, assim como para todos os sistemas multi-modais, a utilização de matrizes de confusão para avaliar e afinar o sistema torna-se um método de particular importância. Nestes sistemas, deve ser determinada uma matriz de confusão para cada um dos *streams* de características individualmente, fazendo-se depois uma análise dessas matrizes para tentar identificar a distribuição dos erros ocorridos, procurando relações ou não, entre elas. Dependendo das conclusões pode-se optar por abordagens específicas à implementação do módulo multi-modal. Por exemplo, identificando que para determinadas palavras (unidades, no caso geral) um dos *streams* apresenta uma elevada discriminação, poder-se-á implementar uma técnica que permita dar relevância na classificação final a esse *stream*, sempre que essas palavras surgirem como hipóteses fortes no classificador.

Neste sistema foram construídas as matrizes confusão dos *stream* áudio e vídeo. Em reconhecimento multi-modal, como se irá ver mais adiante, é possível que um *stream* apresente fracos resultados em determinadas palavras, e que outro *stream* compense essa má performance.

4.2 Resultados

Nas secções seguintes vão ser apresentados os resultados do sistema desenvolvido. A abordagem apresentada acompanha o desenvolvimento que o sistema seguiu.

[1] Através da análise da matriz confusão é possível discriminar entre que palavras o classificador está a trocar.

Vai ser apresentada uma abordagem ao desenvolvimento dos diferentes módulos do sistema, módulo apenas com as características acústicas (*single-stream* áudio), módulo apenas com as características visuais (módulo *single-stream* vídeo) e o módulo *multi-stream* aúdio-visual. Para além de ser apresentada a evolução do desempenho do sistema, também vão sendo apresentadas as diversas decisões tomadas em relação à afinação e implementação do sistema, em especial à decisão sobre os seus parâmetros.

4.2.1 *Single-Stream* Áudio

Sem dúvida que o módulo mais importante de um sistema de reconhecimento de fala audio-visual deverá ser o módulo *single-stream* áudio, desde que operando em condições favoráveis. Um bom desempenho deste módulo permite retirar importantes considerações prévias para o método a seguir na combinação dos dois *streams*, assim como ter uma ideia da taxa de erro para a qual o sistema audio-visual deve evoluir.

O método de paragem do treino do sistema consistiu em atingir uma iteração a partir da qual não se verificavam melhorias nas taxas de reconhecimento, avaliadas sobre o conjunto de frases da base de dados seleccionadas para desenvolvimento ou afinação. O número da iteração considerada óptima varia de acordo com a tarefa de reconhecimento.

O processo de desenvolvimento do módulo *single-stream* áudio começa por considerar um *codebook* de pequena dimensão, 64 gaussianas, de modo a evitar perdas de tempo de processamento nas iterações iniciais. Por outro lado, a utilização inicial de poucas gaussinas[1] traduz-se, geralmente, numa melhor distribuição sobre o espaço de representação, proporcionando uma distribuição equilibrada de pesos por todas as gaussianas. O número de gaussianas utilizado foi aumentando sempre que a taxa de erro estabilizava ou começava a subir. Nesta situação o número de gaussianas era duplicado (modelos das palavras e *codebook*), através de uma técnica de divisão. O processo repetia-se até a capacidade discriminativa dos modelos ser considerada esgotada[2].

No desenvolvimento inicial, os valores do WP e do BW mantiveram-se constantes e iguais a 0 no caso do WP, e 50 para o treino e 100 para a descodificação, no caso do BW, respectivamente.

[1] Quando se utiliza um valor inicial de gaussianas elevado, a distribuição pode não ser tão uniforme, havendo a possibilidade de várias gaussianas ocuparem espaços muito próximos, o que retira capacidade de discirminação ao sistema.

[2] Não apresentar melhoria continuando com o processo de desenvolvimento

A Figura 4.3 e a Tabela 4.1 mostram a evolução da taxa de erro (WER) para as sucessivas iterações durante o processo de treino do *stream* áudio para esta tarefa.

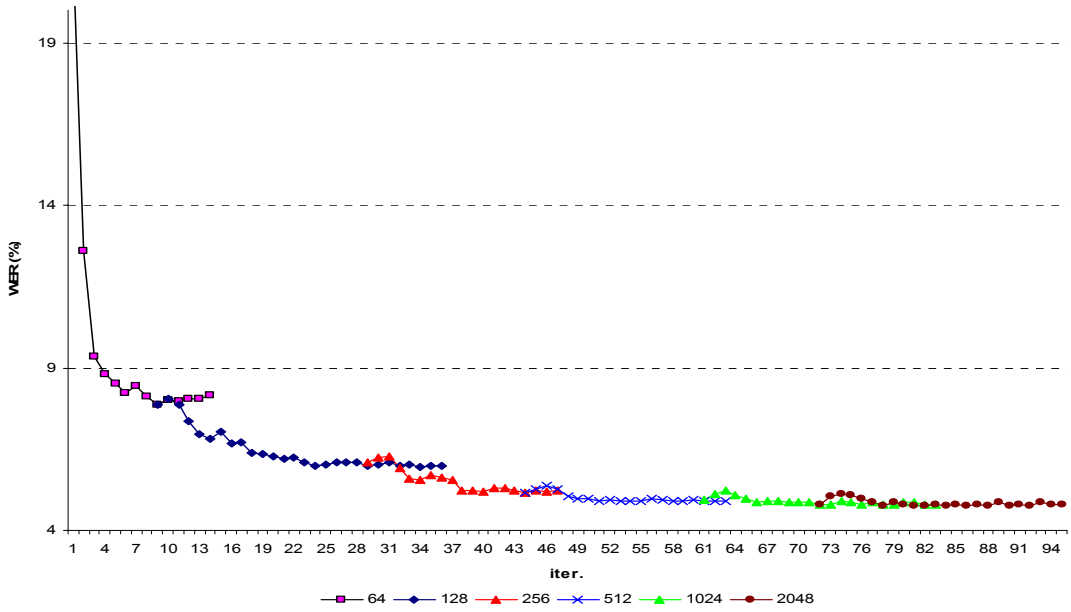


Figura 4.3 - Evolução do WER para o *single-stream* áudio em função da iteração

Tabela 4.1 - Evolução do WER para o *single-stream* áudio (valores mais significativos)

Iteração	Nº Gaussianas	S (%)	D (%)	I (%)	WER (%)
0	64	11,24	2,80	8,04	22,08
8	64	3,44	1,36	3,04	7,84
28	128	2,40	1,28	2,32	6,00
43	256	1,92	1,20	2,04	5,16
60	512	1,80	1,16	1,96	4,92
71	1024	1,76	1,08	1,96	4,80
91	2048	1,72	1,08	1,96	4,76

Da observação da Figura 4.3 e Tabel 4.1 pode-se verificar que a evolução da WER é a esperada. A taxa de erro para os modelos semeados é elevada, 22,08%$_{abs}$, verificando-se uma rápida descida desse valor nas iterações imediatamente seguintes. Mas, rapidamente, ao fim de 8 iterações, o sistema atinge um valor mínimo, iniciando uma ligeira subida dos valores da WER para as iterações imediatamente seguintes. Uma vez que o sistema não apresentava melhorias no seu desempenho, concluiu-se que a evolução do sistema com 64 gaussianas estava esgotada, havendo a necessidade de aumentar esse valor. O valor da WER mínimo atingido foi de 7,84%$_{abs}$. Pode então concluir-se que 64 gaussianas é um valor pequeno para discriminar entre as classes desta aplicação, uma vez que, para o sistema desenvolvido (um só falante e

vocabulário pequeno) o valor atingido fica muito aquém de outros sistemas semelhantes. Procedeu-se então ao aumento do número de gaussianas. O método implemento, de modo a simplificar as alterações consequentes nos algoritmos de treino e descodificação, foi uma simples divisão de cada gaussiana, duplicando assim o seu valor. Passou-se então a ter um sistema a funcionar com 128 gaussianas.

Após a duplicação do número de gaussianas, o valor da taxa de erro sofre um aumento em relação ao valor de paragem para o número de gaussianas anterior. Este aumento inicial é resultado da perda de precisão dos modelos resultante da divisão realizada. Após esse aumento verifica-se um decréscimo do WER para valores inferiores aos atingidos com menos gaussianas. Como seria de esperar, o decréscimo da WER nas iterações iniciais é maior, verificando-se uma diminuição desse valor até estabilizar ou começar a crescer. O valor da WER mínimo atingido foi de 6%. Como para as 64 gaussianas, o valor do WER atingido com 128 gaussianas ficou aquém do esperado. Procedeu-se então ao mesmo procedimento para as iterações seguintes.

Na Figura 4.4 pode-se ver com mais consistência a descida da WER cada vez que se duplica o número de gaussianas. É de realçar que nas primeiras iterações após a duplicação do *codebook* o WER sobe para depois ter uma descida mais acentuada. Essa subida deve-se ao método de divisão utilizado.

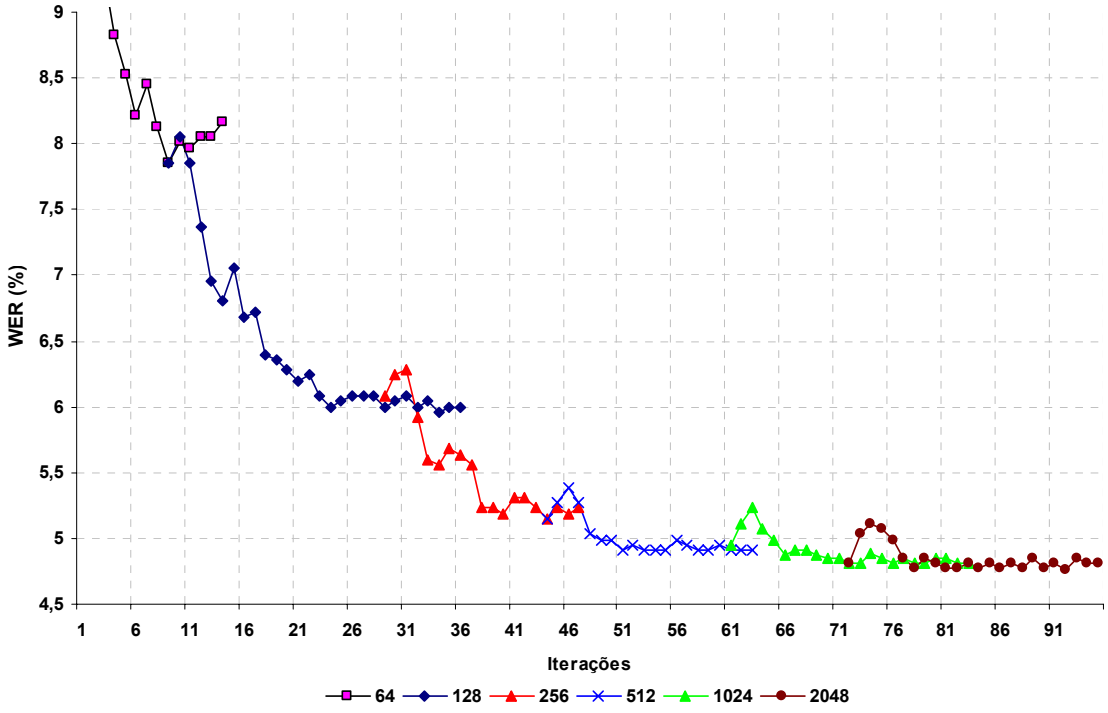


Figura 4.4 - Evolução do WER para o single-stream áudio (WER entre os 4,5% e 9%) em função da iteração

Da análise das figuras apresentadas e da tabela podem ser retiradas uma série de conclusões. Por um lado, o valor mínimo da WER obtido ficou-se por 4,76%$_{abs}$. Este valor é claramente alto para a especificação do sistema, o que levou à necessidade de procurar soluções para baixar este valor, ou, caso contrário, para confirmar que seria um valor aceitável.

Por outro lado, verifica-se que a partir de 512 gaussianas a variação do valor da WER é pouco significativa (4,91%$_{abs}$, 4,81%$_{abs}$ e 4,76%$_{abs}$ para 512, 1024 e 2048 gaussianas respectivamente). A pequena variação verificada no valor do WER é oposta ao crescimento significativo do tempo de processamento requerido pelo sistema para os respectivos números de gaussianas, como apresentado na Tabela 4.2. Deste modo, e como o valor do WER não fica comprometido a partir de 512 gaussianas, e como o tempo de processamento é factor de extrema importância num sistema de reconhecimento de fala[1], ficou logo estabelecido que seria esse valor o número de gaussianas final em posteriores desenvolvimentos do sistema.

Tabela 4.2 - Tempo de processamento vs. WER em função do número de gausseanas

Nº. Gaussianas	**Tempo Processamento**
64	T_{64}
128	~ $2\times T_{64}$
256	~ $4\times T_{64}$
512	~ $8\times T_{64}$
1024	~ $16\times T_{64}$
2048	~ $32\times T_{64}$

Para abordar o problema do valor elevado mínimo para o WER atingido, passou-se à análise dos indicadores disponíveis (D, S, I e matriz de confusão), assim como a uma análise mais específica do tipo de erros ocorridos através da análise individual das frases resultantes confrontadas com as originais. Esta análise mais específica centrou-se também na própria análise do sinal, procurando encontrar condicionantes que normalmente ocorrem nos sistemas de reconhecimento de fala, como sejam a presença de ruídos e outros fenómenos.

Da análise referida, constatou-se que grande parte dos erros se devia principalmente à presença de eventos produzidos pelo orador e não a ruídos exteriores (uma vez que o ambiente na fase de gravação da base de dados estava estável e controlado). Os tipos de condicionantes foram principalmente dois, por um lado o humedecer da boca, e, por outro lado, a respiração (inspiração ou expiração) do falante durante a locução das frases.

[1] Especialmente na implementação do sistema em tempo real

Este tipo de eventos provocava a inserção de novas palavras (palavras pequenas) na fase de descodificação, o que vinha de encontro ao valor obtido para a taxa de inserção, que era claramente superior ao da taxa de apagamentos.

Foram então procuradas soluções. O problema está relacionado com a robustez do sistema, por isso, as condicionantes referidas podem ser classificadas como ruído (que neste caso se prende apenas com o próprio processo de falar, e não com ruídos devido ao ambiente envolvente). Existem diversos métodos para modelar ruídos, sendo habituais as abordagens que se baseiam no nível da modelação do sinal, nomeadamente em alterações ao modelo base das unidades acústicas.

Após a análise da estrutura dos dois tipos de "ruídos" presentes, verificou-se que apresentavam uma estrutura (espectral e temporal) muito simples, podendo ser modelados com poucos estados. Por outro lado, o segundo tipo de eventos referidos aparecia poucas vezes, não permitindo que houvessem elementos suficientes à partida para treinar com precisão um modelo específico. Foi então testada uma primeira abordagem simplista, que se verificou ser suficiente uma vez que eliminou esses problemas.

A abordagem seguida foi a de implementar apenas um modelo para os dois tipos de eventos, com a mesma estrutura dos modelos seguidos para as palavras, ou seja, um modelo esquerda-direita. Esse modelo (Figura 4.5) é composto por dois estados e é semeado e treinado com amostras dos dois tipos de "ruídos" referidos. O modelo foi designado por "click". A esperança de que o modelo conseguisse ser robusto, de modo a poder classificar correctamente ambos "ruídos" verificou-se, tendo resultado numa clara subida do desempenho do sistema, não tendo sido necessário o desenvolvimento de uma abordagem mais complexa.

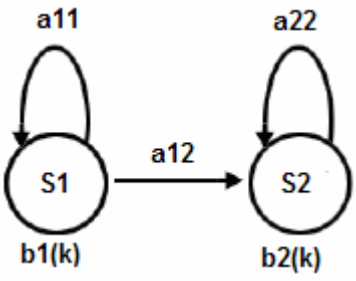
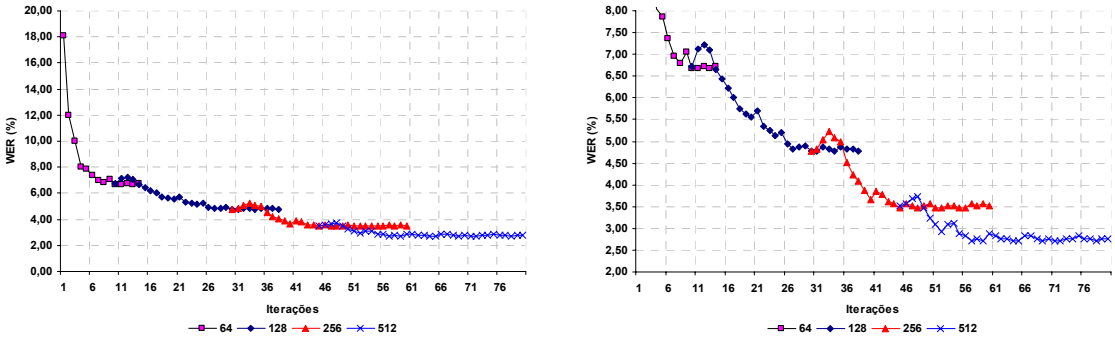


Figura 4.5 - Estrutura do modelo HMM para o "click"

Após adaptação do sistema para mais um modelo, foi repetido todo o processo apresentado, mas apenas até o valor do WER estabilizar usando 512 gaussianas. Os resultados obtidos são apresentados na Figura 4.6, tendo sido atingido o valor mínimo do WER de 2,72%, na iteração 63. Na Tabela 4.3 são apresentados os valores mais significativos da evolução do WER.



a) b)

Figura 4.6 - Evolução do WER para o single-stream áudio já com o modelo do click. a) Visualização da evolução geral; b) Visualização da evolução para valores entre 2% e 8%.

Tabela 4.3 - Evolução do WER para o single-stream áudio (valores mais significativos) em função da iteração

Iteração	Nº Gaussianas	S (%)	D (%)	I (%)	WER (%)
0	64	8,88	4,32	4,88	18,08
9	64	2,28	2,18	2,22	6,68
29	128	1,52	1,64	1,64	4,80
44	256	1,12	1,16	1,20	3,48
63	512	0,84	0,92	0,96	2,72

De seguida foi abordada a questão da modelação linguística do sistema. Para lidar com esta questão foram introduzidos no sistema, numa fase inicial uma gramática *Word Pair*, e posteriormente uma gramática *Bigram* .

Na Tabela 4.4 apresentam-se os diversos resultados obtidos. Verificou-se um aumento do desempenho do sistema quer, primeiro, com a utilização da *Word Pair*, quer, posteriormente, com a utilização da *Bigram*. Esta diminuição era esperável uma vez que com a sua inclusão, a perplexidade do sistema reduziu significativamente.

Tabela 4.4 - Valores da WER para o single-stream áudio para as várias estruturas do sistema

Medida de avaliação	Sem Gramática		Com gramática	
	Sem Click	Com click	Word Pair	Bigram
SA (%)	25,42	64,41	74,58	86,44
S (%)	1,72	0,84	0,84	0,76
D (%)	1,08	0,92	0,56	0,24
I (%)	1,96	0,96	0,64	0,12
WER (%)	4,76	2,72	2,04	1,12

Da análise da tabela, verifica-se que existe um equilíbrio entre os valores da taxa de palavras inseridas e apagadas. Mesmo assim, foram realizados vários testes para diferentes valores de WP e, como seria de esperar, não se verificou nenhuma melhoria do valor da WER.

O valor mínimo da WER, obtido com o conjunto de frases da base de dados para desenvolvimento, com gramática *Bigram*, foi de 1,12%. Este valor pode ser considerado um valor aceitável, dada a especificação do sistema.

Antes de se passar para a avaliação do sistema *single-stream* vídeo, e dado o desempenho do sistema, só por $1{,}12\%_{abs}$ de erro não parece relevante a utilização das características visuais, uma vez que pouco poderia melhorar no desempenho do sistema. Todavia, um dos objectivos do sistema é provar que o *stream* vídeo pode ser utilizado como uma metodologia auxiliar de modo a conferir maior robustez ao sistema de reconhecimento, nomeadamente quando as condições acústicas envolventes forem algo adversas. A relevância das características visuais num ambiente acusticamente adverso assenta na sua independência dos ruídos acústicos.

Às frases originais foi adicionado ruído AWGN de modo a baixar a SNR, tendo sido obtidas as SNR de 19dB, 14dB, 9dB e 4dB. Foram realizadas duas avaliações. Por um lado, as frases de desenvolvimento com as SNR referidas foram avaliadas pelos modelos originais, treinados com a melhor relação SNR, obtidos por afinação do sistema em função das frases gravadas. Por outro lado, foi repetido todo o procedimento (treino e afinação) de determinação dos modelos (modelos diferentes para diferentes SNR) que permitem obter menores valores de WER, utilizando a base de dados com os diferentes valores de SNR.

Como seria de esperar, o valor da WER utilizando os modelos determinados com as próprias frases afectadas pelo ruído adicional apresentaram valores mais baixos de WER, ou seja, melhor desempenho, como está apresentado na Figura 4.7. Para melhor análise da diferença entre as duas situações apresenta-se a Tabela 4.5.

Tabela 4.5 - Evolução do WER em função do SNR para os modelos limpos e modelos ruidosos (Sistema com *Bigram*)

SNR (dB)	**WER (%)**	
	Modelos Limpos	**Modelos Ruidosos**
24	1,12	1,12
19	3,56	2,96
14	20,48	18,24
9	38,96	32,52
4	61,44	50,16

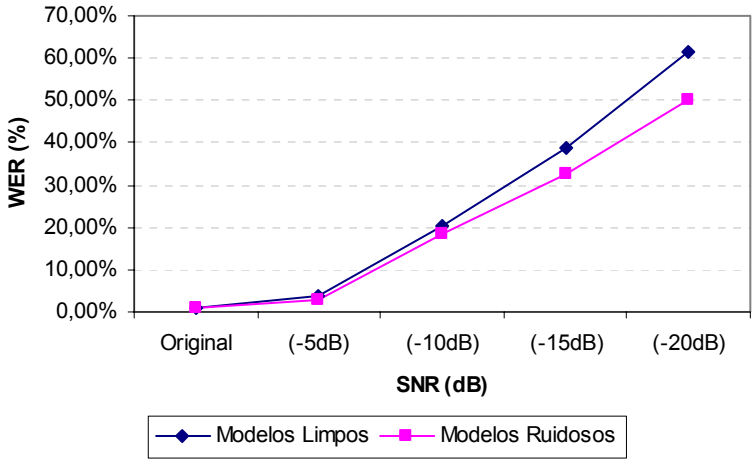


Figura 4.7 - Variação do WER em função do SNR para os modelos limpos e modelos ruidosos (Sistema com *Bigram*)

Nos resultados que se apresentam a seguir para o sistema audio-visual, sempre que aparecer a WER em função do SNR, os modelos utilizados serão os modelos acústicos afinados com ruído, uma vez que são os que supostamente menos desvirtuam o que acontece numa aplicação real (os sistemas a funcionar em tempo real geralmente utilizam a adaptação dos modelos, o que lhes confere maior robustez).

Uma das consequências que seria de esperar era uma diferença maior entre o desempenho do sistema com os modelos limpos e com os modelos ruidosos. Essa diferença foi atenuada uma vez que se utilizou a *Bigram*, que baixa a perplexidade do sistema, tendo um reflexo mais evidente em condições mais adversas.

Da análise dos resultados obtidos, se verifica claramente que com a diminuição da SNR o desempenho do sistema baixa claramente, justificando-se a necessidade de robustecer o sistema, o que vai ser conseguido através da inclusão das características visuais.

4.2.2 *Single-Stream* Video

A estrutura do sistema *single-stream* vídeo é a mesma do sistema áudio. Deste modo, após as adaptações nos códigos dos algoritmos para o *stream* áudio, procedeu-se ao desenvolvimento e avaliação do sistema visual.

O procedimento seguido foi análogo, tendo sido realizados uma série de testes para verificar se algumas das decisões tomadas para o sistema acústico se poderiam generalizar para o sistema visual. Deste modo, após uma pequena série de testes, verificou-se que a utilização do modelo "click" trazia um melhor desempenho do sistema visual. Não foi possível verificar se a melhoria

resultante da introdução desse modelo é tão eficaz como no sistema áudio, uma vez o número de erros é muito maior, não sendo rápida uma análise eficaz da sua utilização. Também foram realizados testes com mais de 512 gaussianas, não resultando numa diminuição do WER, igual ao que aconteceu no sistema acústico, apresentado no sistema visual uma maior instabilidade nos seus valores. Ficou então estabelecido que o número máximo de gaussianas seria de 512 e que seria utilizado o modelo auxiliar "click".

Por outro lado, os valores do WP e do BW mantiveram-se constantes e iguais a 0 no caso do WP, e 150 para o treino e 200 para a descodificação no caso do BW, respectivamente. O aumento deste valor foi decidido após as primeiras iterações do sistema, uma vez que as características visuais das frases não se conseguiam prender aos modelos das palavras considerando um BW menor.

Os principais resultados obtidos podem ser observados na Tabela 4.6. Embora a evolução relativa dos valores da WER no caso visual apresentem a mesma estrutura que no caso acústico, esses valores são muito mais elevados, encontrando-se na gama de valores entre os 60% e os 30%. O valor inicial da WER para os modelos semeados e utilizando 64 gaussianas e sem gramática é de 72,36%$_{abs}$, atingindo o valor mínimo de 49%$_{abs}$ com 512 gaussianas, sem utilizar qualquer gramática. Como no sistema *single-stream* áudio, para diminuir esse valor foram inseridas a *Word Pair* e a *Bigram* tendo o sistema visual obtido os valores para a WER de 42,12%$_{abs}$ e 36,88%$_{abs}$, respectivamente.

Tabela 4.6 - Valores da WER para o *single-stream* visual para as várias estruturas do sistema

Medida de avaliação	**Sem Gramática**	**Com gramática**	
	Com Click	**Word Pair**	**Bigram**
SA (%)	0,85	8,62	13,56
S (%)	29,46	24,16	18,84
D (%)	6,62	7,16	7,62
I (%)	12,92	10,80	10,42
WER (%)	49,00	42,12	36,88

Uma análise mais cuidadosa à Tabela 4.6 permite verificar que, neste caso, poderá ser pertinente o ajustamento do valor do WP. Embora a relação existente entre o número de palavras inseridas e apagadas seja semelhante ao caso acústico, neste caso, esses valores são muito superiores, havendo a possibilidade que tal ajustamento se traduzir num ganho para o sistema.

Nesse sentido foi realizada uma série de testes para diferentes valores de WP. Nesses testes o WP tomou valores entre -1 e 1, tendo-se verificado uma alteração pouco significativa do desempenho do sistema. O melhor resultado foi obtido para um WP de -0,7 tendo se conseguido um ganho de $5\%_{rel}$ (WER tomou o valor de $36{,}68\%_{abs}$), que se traduziu no acerto de mais 5 palavras entre 922 erradas com o WP inicial (igual a 0). Dos testes realizados observou-se que quando o WP assume um valor negativo, o sistema consegue um desempenho melhor, resultante da aproximação entre os valores do número de palavras inseridas e apagadas. Quando o WP tomou valores positivos, a diferença entre o número de palavras inseridas e apagadas aumentou, assim como o valor do WER, resultando num desempenho inferior do sistema.

Apesar da variação do WP ter resultado num ganho para o sistema esse valor não foi considerável de modo a poder assumi-lo no desenvolvimento do sistema multi-stream, razão pela qual se decidiu continuar a utilizar como base de desenvolvimento o valor que lhe foi inicialmente atribuído.

Os valores obtidos para o sistema visual, apesar de um pouco elevados, dado tratar-se de um sistema de um só orador, apresentam-se relativamente próximo dos obtidos para sistemas semelhantes, levando desde logo a perspectivar-se um ganho no desempenho do sistema numa abordagem conjunta dos *streams* acústico e visual.

4.2.2.1 Testes de Leitura Labial

Antes de se passar para a implementação e análise do sistema *multi-stream* decidiu-se tentar obter um indicador para os resultados obtidos no sistema visual. Ao contrário do sistema acústico, onde já existem muitos estudos que permitem validar os resultados obtidos, para o *stream* vídeo, esses estudos são muito menos frequentes, especialmente devido às condicionantes relativas à aquisição e preparação de um base de dados audio-visual, tendo sido encontrado poucos sistemas semelhantes ao apresentado. Deste modo, não pode ser assumida uma gama de valores restrita de referência para a WER caminhar.

O teste projectado consistiu, como em muitos outros testes realizados na área do processamento da fala[1], num confronto homem-máquina. O teste consistiu em colocar pessoas, com conhecimentos e capacidades específicas, a realizar o reconhecimento do conteúdo linguístico das mesmas frases que foram classificadas pelo sistema visual automático. Essa classificação consistia na transcrição ortográfica das frases reconhecidas através da observação do movimento labial do orador durante a locução das frases, sem recurso à componente

[1] Especialmente em testes realizados no âmbito da síntese da fala (tese de perceptibilidade, etc.)

acústica. É preciso não esquecer que a zona labial corresponde à ROI utilizada na modelação visual.

O teste foi realizado por duas pessoas devidamente preparadas e com conhecimentos de leitura labial. O primeiro indivíduo, D. O., de 23 anos, é estudante finalista do curso de Matemática Aplicada na Faculdade de Ciências da Universidade do Porto. Os seus conhecimentos de leitura labial estão directamente relacionados com uma deficiência congénita (dificuldades de audição e locução) e remontam aos seus 3 anos de idade, tendo frequentado, por um período de 2 anos, um colégio especial onde aprendeu a leitura labial. O segundo indivíduo, M. F., de 34 anos, é licenciado e exerce funções de educador em instituições para problemas com deficiências. Os seus conhecimentos remontam principalmente aos últimos 10 anos, quando iniciou a sua vida profissional, com a particularidade de na sua família existirem 2 pessoas surdas-mudas, o que desde cedo levou a aprender a leitura labial.

O teste iniciava-se, tal como o sistema implementado, através de um treino, que neste caso pretendia familiarizar os dois participantes com as características específicas do movimento labial do orador durante a fala. Após o treino, e para uma comparação mais próxima possível com o sistema, foram colocadas as mesmas frases utilizadas no desenvolvimento. Foi pedido aos avaliadores que se tentassem abstrair o máximo possível do conteúdo semântico das frases, de modo a evitar que fossem realizadas correcções devido aos seus conhecimentos próprios da língua Portuguesa, transcrevendo apenas as palavras que realmente conseguiam identificar. É natural que apesar de toda a abstracção possível por sua parte, os conhecimentos inerentes da estrutura gramatical da língua, em especial da estrutura das frases do sistema desenvolvido (calculadora) levam a que involuntariamente palavras possam ser eliminadas em certas fases da classificação. Por exemplo, mesmo que durante a classificação possa surgir a possibilidade de ocorrer uma palavra que é impossível, instintivamente, será eliminada e procurada uma nova solução no conjunto de palavras do vocabulário que gramaticalmente sejam possíveis. Pode então ser considerado que os resultados que se apresentam na Tabela 4.7 não correspondem à situação do sistema sem gramática, mas também não correspondem a nenhuma das duas situações com *Word Pair* ou *Bigram*.

Tabela 4.7 - Resultados obtidos das experiências de leitura labial

Medida de avaliação	**D. P.**	**M. F.**
SA (%)	26,62	28,12
S (%)	9,10	18,78
D (%)	20,30	12,26
I (%)	1,88	4,68
WER (%)	31,28	28,12

Da análise da Tabela 4.7 verifica-se que os resultados obtidos são relativamente semelhantes para as dois avaliadores o que confere, à partida, alguma confiança aos resultados. Os valores apresentados encontram-se melhores que os dos resultados obtidos para o sistema, mesmo quando este usa a *Bigram* na descodificação. Uma primeira conclusão a retirar é que o sistema apresenta um resultado um pouco abaixo do que poderia ser atingido, uma vez que as frases de desenvolvimento foram utilizadas para afinação dos modelos do sistema e no entanto, tem uma taxa de sucesso inferior à de qualquer um dos avaliadores. Outra evidência dos resultados apresentados reside no facto que, ao contrário do sistema que introduzia mais palavras (dividia palavras de grande dimensão inserindo palavras mais pequenas) do que apagava, qualquer um dos avaliadores apagava muito mais palavras (quase sempre palavras pequenas, especialmente do grupo dos conectores) do que inseria (a taxa de inserções apresentada foi muito baixa). Pode-se então concluir que os avaliadores conseguiram com bastante mais precisão determinar as fronteiras temporais das palavras, em especial as de grande dimensão, falhando principalmente em palavras de ligação, "e", "de", "a". É preciso ter em consideração que os avaliadores têm conhecimento das palavras, da estrutura das frases, e do seu contexto na aplicação[1], o que facilita a tarefa de identificação das fronteiras das palavras, não se sabendo se teriam a mesma facilidade nessa identificação se não tivessem a mesma familiaridade com o contexto da aplicação. Outra observação que é importante referir é que para os avaliadores, os movimentos labiais que não pertenciam a nenhuma palavras ("ruídos") eram claramente identificados.

O teste realizado pelo primeiro indivíduo, D. O., foi totalmente supervisionado, tendo-se por repetidas vezes verificado a tentativa espontânea de correcção de palavras, através do conhecimento inerente da estrutura gramatical da língua Portuguesa, em geral, e do sistema (uma calculadora) mais especificamente. Por diversas vezes notou-se uma tentativa de colocar palavras de lado, ou substitui-las por outras que encaixavam nas regras gramaticais possíveis. Por este motivo, e para uma análise posterior, após a classificação de cada frase pelo D. O., foi anotada uma segunda vez cada frase mas agora com possibilidade de aplicar todos os conhecimentos referidos. Como seria de esperar, os indicadores do desempenho tiveram uma melhoria significativa, baixando o valor da WER para 21,24%, como se pode verificar na Tabela 4.8. É importante referir que este valor é obtido através de um nível de conhecimento que é muito difícil de implementar em sistemas de reconhecimento.

[1]Fornece uma indicação dos benefícios inerentes à inclusão do modelo linguístico no sistema de reconhecimento automático de fala

Tabela 4.8 - Resultados obtidos das experiências de leitura labial (aplicando os conhecimentos linguísticos)

Medida de avaliação	**D. P.**
SA (%)	54,68
S (%)	8,23
D (%)	10,58
I (%)	2,43
WER (%)	21,24

De uma análise mais pormenorizada dos erros cometidos pelos dois avaliadores, mesmo no teste que incluiu o conhecimento de toda a estrutura sintáctica e gramatical, verificou-se que os erros de substituição (responsáveis por mais de metade do valor da WER) se concentravam quase exclusivamente em três ou quatro palavras, "sessenta" e "setenta", "cinco" e "oito". Se não for considerado o último teste referido, a palavra "de" era bastantes vezes substituída por "e", e o próprio "e" era muitas vezes suprimido.

Para finalizar, referir que este estudo apresentado, dá uma indicação importante em relação aos resultados esperados para o sistema visual. É importante referir que os resultados obtidos não pode ser generalizados, nem assumir conclusões finais, uma vez que o número de testes foi reduzido. No entanto, dada a proximidade dos resultados obtidos para os dois avaliadores, estes testes podem ser levados em consideração como uma indicação relevante.

4.2.3 *Multi-Stream* Síncrono

Depois de obtidos os modelos com melhor desempenho, quer para as características acústicas, para todas as SNR, quer para as características visuais, foi desenvolvido o sistema para combinar essas características.

A metodologia seguida, uma vez que se tinham determinado os modelos acústicos e visuais com melhor desempenho separadamente, foi apenas o de desenvolver um classificar *multi-stream*, que realizasse a combinação dos dois *streams* de características.

Numa abordagem exploratória, foi implementado um classificador síncrono, em que a contribuição de cada *stream* para a classificação final é realizada através de um parâmetro, "prob". O "prob" não é mais do que um valor entre 0 e 1 que determina qual o peso de cada *stream*. Se o peso do *stream* áudio valer "prob", o peso do *stream* visual será "1-prob". O objectivo é determinar para cada SNR, qual é o valor que o "prob" deverá tomar de modo a minimizar o valor da WER.

Nos resultados que vão ser apresentados, quando o "prob" assume o valor 1 o descodificador apenas considera o *stream* acústico e quando assume o valor 0 apenas considera o *stream* vídeo.

Na afinação do valor "prob" foram utilizadas as mesmas frases que foram utilizadas para desenvolvimento dos sistemas single-stream, e os valores do WP e do BW mantiveram-se constantes e iguais a 0 no caso do WP, e 150 para o treino e 200 para a descodificação no caso do BW, respectivamente.

O processo iniciou-se para os modelos acústicos gerados através das frases originais. Numa primeira fase eram testados valores de "prob" entre 0 e 1 com espaçamento de 0,1 entre cada dois valores consecutivos. Seguidamente analisava-se entre que gama de valores se poderia encontrar o valor óptimo, repetindo-se os testes dentro dessa gama, para intervalos de análise mais pequenos (de 0,025 cada).

Para os modelos originais, a evolução da WER durante a afinação do "prob" é apresentada na Figura 4.8. Na Tabela 4.9 é apresentada a primeira fase de análise, em que se determina que o valor óptimo se encontra entre os 0,7 e 1, e na Figura 4.9 e Tabela 4.10 apresenta-se a procura mais fina do valor óptimo dentro da gama previamente determinada, tendo resultado no valor de 0,85. A pesquisa pretende obter o melhor valor dentro de um conjunto de valores definidos por um determinado critério (espaçamento fixo entre valores a testar). O valor óptimo era determinado após a segunda pesquisa (espaçamento de 0,025 entre cada dois valores) uma vez que não é esperado um ganho que justifique o tempo requerido, e por outro lado, este valor está claramente condicionado ao grupo de frases que estão a ser testadas, podendo esse valor ser diferente para outro conjunto de frases. O valor da WER mínimo atingido foi de 0,92%$_{abs}$, ou seja, um ganho aproximado de 20%$_{rel}$ em relação ao *stream* áudio (o mais forte para esta SNR).

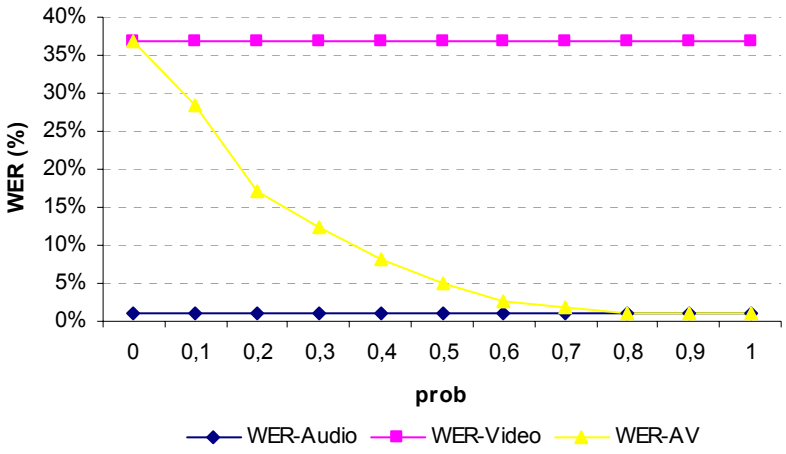


Figura 4.8 - Variação do WER em função do parâmetro "prob" para o SNR original

Tabela 4.9 - Variação do WER para o SNR original e "prob" entre 0 e 1

Com Bigram e SNR 24dB			
Prob	WER-Audio	WER-Video	WER-AV
0	1,12%	36,88%	36,88%
0,1	1,12%	36,88%	28,52%
0,2	1,12%	36,88%	17,04%
0,3	1,12%	36,88%	12,36%
0,4	1,12%	36,88%	8,08%
0,5	1,12%	36,88%	4,88%
0,6	1,12%	36,88%	2,76%
0,7	1,12%	36,88%	1,88%
0,8	1,12%	36,88%	1,04%
0,9	1,12%	36,88%	1,12%

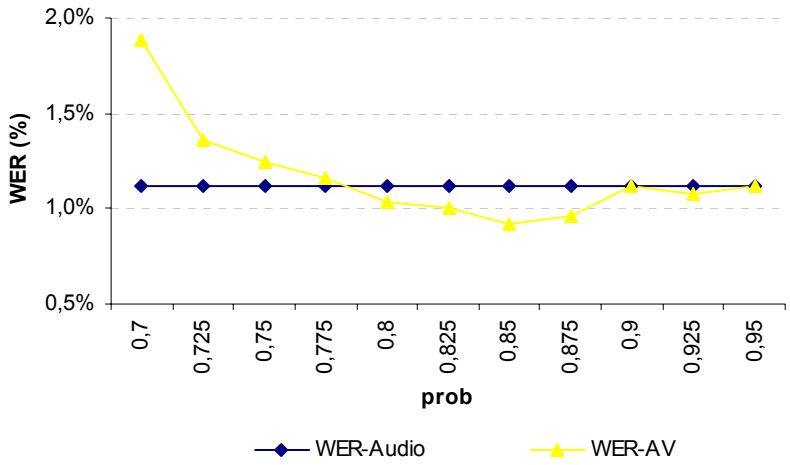


Figura 4.9 - Variação do WER em função do parâmetro "prob" (entre 0,7 e 1) para um SNR original

Tabela 4.10 - Variação do WER para um SNR original e "prob"entre 0,7 e

Com Bigram e SNR 24dB			
Prob	WER-Audio	WER-Video	WER-Final
0,7	1,12%	36,88%	1,88%
0,725	1,12%	36,88%	1,36%
0,75	1,12%	36,88%	1,24%
0,775	1,12%	36,88%	1,16%
0,8	1,12%	36,88%	1,04%
0,825	1,12%	36,88%	1,00%
0,85	1,12%	36,88%	0,92%
0,875	1,12%	36,88%	0,96%
0,9	1,12%	36,88%	1,12%
0,925	1,12%	36,88%	1,08%
0,95	1,12%	36,88%	1,12%

O processo repetiu-se para todas as SNR tendo sido obtidos os valores de "prob" óptimos que se apresentam na Tabela 4.11. Como era de esperar, o valor do "prob" é tanto menor quanto menor for a SNR, isto porque, maior é a taxa de erros do *stream* acústico, logo, maior deverá ser a contribuição do *stream* visual. As tabelas e gráficos parciais de afinação do parâmetro para as diferentes SNR calculadas são apresentados em anexo (ver Anexo C).

Tabela 4.11 - Valores óptimos do "prob" e WER atingidas para diferentes SNR

SNR (dB)	prob	WER-AV (%)
24	0,850	0,92
19	0,775	2,40
14	0,650	12,24
9	0,550	24,04
4	0,500	33,56

Outra das conclusões que se pode retirar dos resultados obtidos para os valores óptimos calculados para diferentes SNR e apresentados na Tabela 4.10 é o ganho do sistema *multi-stream* em relação ao sistema *single-stream* acústico. Esse ganho é determinado através da expressão (4.10).

$$Ganho = \frac{WER(Single-stream\ audio) - WER(Multi-stream)}{WER(Single-stream\ audio)} \quad (\%_{rel}) \qquad (4.10)$$

Na Figura 4.10 é apresentado um gráfico com a evolução das WER durante a afinação do "prob" para todas as SNR calculadas. Da análise verifica-se que com a diminuição do valor da SNR, o valor mínimo da WER é atingido para valores menores do "prob", ou seja, verifica-se a deslocação de valor óptimo no sentido de dar mais peso à componente visual. Outra das observações que podem ser retiradas é que quanto menor for a SNR maior é a oscilação relativa da WER para mesmos valores de *"prob"*.

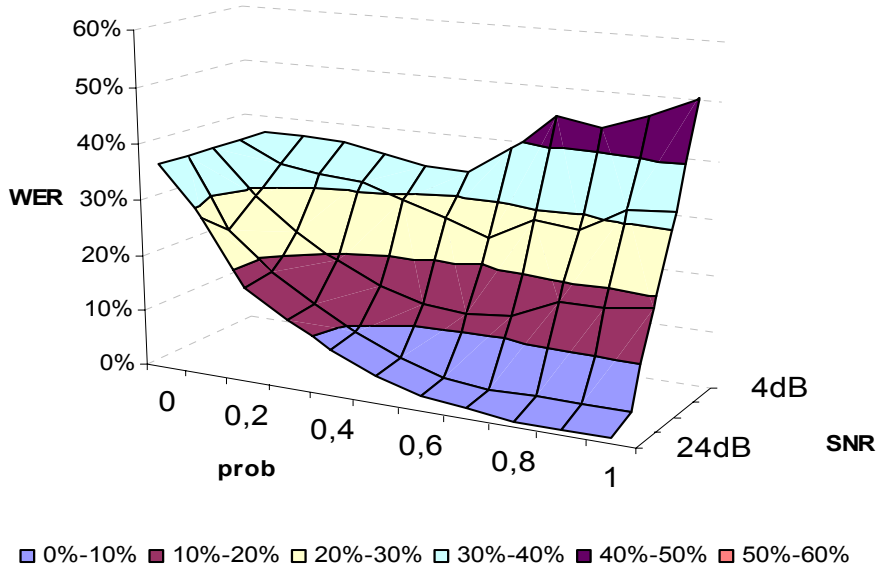


Figura 4.10 - Variação do WER em função do parâmetro "prob" para um SNR de

4.3 Avaliação Final do Sistema

Na secção anterior apresentaram-se os resultados do sistema durante o seu desenvolvimento. Os modelos das palavras finais determinados, assim como os valores de determinados parâmetros estão afinados para um conjunto de 119 frases seleccionadas exclusivamente para o efeito podendo ter sido essa afinação enviesada para determinadas características presentes nesse conjunto.

Para validar todo o desenvolvimento, foi criado uma conjunto de frases, pertencentes ao grupo de teste final ou validação, composta por 118 frases que o sistema desconhece, ou seja, não foram utilizadas nem na fase de treino, nem na fase de desenvolvimentos.

Os resultados do sistema *single-stream* áudio para o conjunto de teste final são apresentados na Tabela 4.12. Podemos verificar o resultados do WER para o sistema sem gramática, com *Word Pair* e com *Bigram*, constatando-se como seria de esperar um aumento da WER. A variação máxima ocorre com a *Bigram* e é de $64\%_{rel}$, que apesar de ser um valor considerável em termos relativos, em termos absolutos se traduz num aumento pouco significativo uma vez que não chega a $1\%_{abs}$. O aumento verificado esta directamente relacionado com o teste ter sido realizado com frases que o sistema não conhecia. Deste modo, algum modelo que possa ter ficado treinado com maior/menor precisão poderá levar a erros por se a variabilidade das amostras entre o conjunto de treino, afinação e teste final for elevada.

Tabela 4.12 - Valores da WER para o *single-stream* áudio para as várias estruturas do sistema (conjunto de teste)

Taxa	Sem Gramática		Com gramática	
	Sem Click	Com click	Word Pair	Bigram
SA (%)	13,56	47,46	62,71	68,64
S (%)	2,86	1,46	1,46	1,28
D (%)	1,32	0,70	0,74	0,32
I (%)	1,74	1,08	0,96	0,24
WER (%)	5,92	3,24	3,16	1,84

De seguida, o sistema *single-stream* áudio foi avaliado para diferentes SNR. Tal como na fase de desenvolvimento, as frases de validação foram testadas com os modelos acústicos desenvolvidos com ruído. Os resultados obtidos são apresentados na Tabela 4.13 e verifica-se o aumento do valor da WER em função da diminuição do SNR.

Tabela 4.13 - Valores óptimos do "prob" e WER atingidas para diferentes SNR (conjunto de teste) do *sistema single-stream* áudio

SNR (dB)	WER (%)	
	Modelos Limpos	Modelos Ruidosos
24	1,84%	1,84%
19	4,12%	3,08%
14	24,96%	19,68%
9	40,12%	34,20%
4	68,88%	50,96%

Os resultados do *single-stream* vídeo são apresentados na Tabela 4.14. Neste *stream*, as variações relativas apresentadas são menores que no sistema *single-stream* áudio mas, verificou-se uma diminuição do seu desempenho.

Tabela 4. 14 - Valores da WER para o *single-stream* vídeo para as várias estruturas do sistema (conjunto de teste)

Taxa	Sem Gramática	Com gramática	
	Com Click	Word Pair	Bigram
SA (%)	1,70	10,17	38,44
S (%)	31,56	25,56	21,04
D (%)	6,84	8,12	6,16
I (%)	14,48	11,04	11,24
WER (%)	52,88	44,72	38,44

Procedeu-se finalmente à validação do sistema *multi-stream* síncrono. Na Figura 4.11 pode verificar-se a evolução da WER em função do SNR, para os valores de "prob" afinados no conjunto de desenvolvimento. Esses valores já foram apresentados na Tabela 4.11. Tal como nos sistemas *single-stream*, verifica-se um aumento relativo do valor da WER, não sendo crítico no que diz respeito ao desempenho final do sistema. Este aumento do WER leva a pensar que os conjuntos de treino, desenvolvimento (ou afinação) e teste final (ou validação) não apresentam uma divisão de dados uniforme, levando a que alguns modelos ao serem afinados num conjunto, percam alguma da sua precisão no outro. Este problema deve-se à definição inicial das frases da base de dados, uma vez que se tentou conseguir concentrar no menor número possível de frases, um número de amostras aceitável, por palavra do vocabulário, para o desenvolvimento e teste do sistema. Deste modo, pode ocorrer que as amostras para determinadas palavras não sejam distribuídas de modo equilibrado pelos vários sub-conjutos formados.

Por outro lado, continuou a verificar-se um benefício da utilização das características visuais em conjunto com as características acústicas, mesmo quando a SNR é máxima. Este resultado prova que o desempenho do sistema visual, apesar de ser baixo, acerta em palavras que o *stream* áudio falha, melhorando o desempenho do sistema quando os dois *streams* são combinados correctamente. Esta conclusão pode ser retirada da análise das matrizes de confusão para os *streams* áudio e visual.

Verificou-se, como esperado que a matriz de confusão do *stream* áudio apresenta uma forma muito bem definida, apresentando os seus valores concentrados na diagonal principal, ou seja, o classificador para este *stream* apresenta um bom desempenho. Por outro lado, como o sistema visual apresenta uma taxa de erro superior, os valores de matriz confusão estão mais espalhados, encontrando-se picos em valores fora da diagonal principal. Por exemplo, as palavras "dois" e "dez" apresentam alguma confusão na componente acústica, mesmo com SNR máximo, não se tendo verificado nenhum erro entre essas duas palavras na componente visual. A inclusão do *stream* visual sempre que o sistema acústico se deparar com palavras acusticamente confusas, poderá permitir facilmente resolver essa situação. Através da análise da matriz confusão audio-visual verificou-se uma suavização de alguns picos de "confusão" verificados para o stream acústico. Verificou-se ainda que as palavras "sessenta" e "setenta" tiveram baixas taxa de acerto tanto no *stream* áudio como no *stream* vídeo, verificando-se em 90% das situações a substituição da palavra "setenta" por sessenta (essa confusão também se verificou no teste de leitura labial). A identificação destas situações é fundamental para o desenvolvimento e afinação do sistema uma vez que as palavras se apresentaram confusas nas duas abordagens. Uma primeira análise poderia consistir em incluir no sistema uns indicadores de modo a verificar as distâncias entre os modelos das duas palavras e qual o nível de confiança dos modelos em geral.

Em ambientes acusticamente desfavoráveis, a inclusão do *stream* visual é ainda mais evidente, melhorando o desempenho do sistema, quando menor for a SNR. Na Tabela 4.15 apresenta-se os resultados do sistema *multi-stream* audio-visual em que o peso relativo de cada *stream*, para diferentes SNR foi o determinado no conjunto de desenvolvimento.

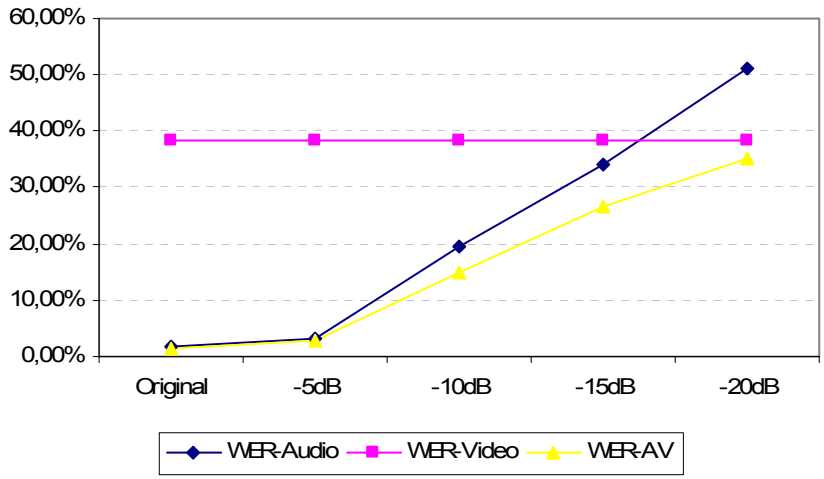


Figura 4.11 - Variação do WER em função do SNR no conjunto de teste

Tabela 4. 15 - Valores da WER para o *multi-stream* em função dos "prob" óptimos (conjunto de teste)

SNR (dB)	prob	WER-AUDIO (%)	WER-AV (%)
24	0,850	1,84%	1,44%
19	0,775	3,08%	2,38%
14	0,650	19,68%	14,80%
9	0,550	34,20%	26,64%
4	0,500	50,96%	35,16%

Da análise da Figura 4.11 e Tabela 4.15 pode-se constatar que o sistema audio-visual apresentou um ganho de 0.40%$_{abs}$, 0.7%$_{abs}$, 4.88%$_{abs}$, 7,56%$_{abs}$ e 15.80%$_{abs}$ para 24dB, 19dB, 14dB, 9dB e 4dB, respectivamente. Em termos de SNR verifica-se que entre os 10% e os 20% o sistema áudio-sisual tem um ganho absoluto por volta de 3 dB.

Deste modo fica provado que combinação dos dois *streams* se traduz num melhor desempenho para o sistema, e que o contributo da componente visual é mais significativa quando o valor da SNR acústica for reduzido.

Uma análise importante é a avaliação do sistema em função do desempenho da aplicação para qual foi desenvolvido, ou seja, operar uma calculadora científica. Tão importante como a

medida do valor de palavras erradas é saber qual é a percentagem de frases acertadas e que levam à uma resposta correcta. A importância da abordagem audio-visual neste ponto torna-se mais evidente quando o sistema está a funcionar em condições favoráveis de SNR. Embora o ganho verificado na abordagem audio-visual seja maior quanto menor for a SNR, se se pensar no funcionamento real (prático) da aplicação, ter um sistema com uma WER superior a 15% não tornará a aplicação interessante. Como se pode verificar, o ganho de $0,4\%_{abs}$ (no conjunto de teste final) nas melhores condições de SNR é bastante significativo pois, nessa situação o sistema acústico erra sensivelmente uma palavra por frase. Deste modo, cada palavra que seja errada apenas com a componente acústica e que seja acertada na abordagem *multi-stream* traduz-se numa frase que passa a estar completamente acertada. O ganho $0,4\%_{abs}$ verificado traduziu-se em mais 10 palavras que acertadas, verificando-se também mais 10 frases completamente acertadas. Deste modo, a taxa de farses acertadas, SA, passou de $68,64\%_{abs}$ para 77,12%, ou seja, $15\%_{rel.}$

CAPÍTULO 5

Conclusões e Trabalho Futuro

Ao longo deste trabalho estudou-se as vantagens de integrar a componente visual nos reconhecedores automáticos de fala contínua *standard*. O trabalho foi iniciado com o desenvolvimento de um sistema *single-stream* áudio de modo a avaliar o desempenho do reconhecimento de fala *standard* para a tarefa implementada. Seguidamente foi realizado um estudo à componente visual da fala, que conduziu ao desenvolvimento de um sistema *single-stream* vídeo. Os resultados apresentados permitem mostrar que a combinação de *streams* áudio e vídeo constitui uma abordagem eficiente em aplicações de reconhecimento de fala contínua, em Português Europeu, especialmente em situações com baixa relação sinal-ruído.

5.1. Conclusões

O objectivo deste trabalho consistia no desenvolvimento de um sistema de reconhecimento de fala audio-visual para o Português Europeu. Os resultados experimentais apresentados são obtidos numa tarefa de reconhecimento de fala contínua dependente do orador e com um vocabulário inicial de 68 palavras de frases destinadas a operar uma máquina de calcular científica[1].

[1] As frases correspondem a expressões matemáticas

O sistema foi desenvolvido utilizando a base de dados LPFAV2, criada no decorrer desta dissertação. Pelas suas características, esta base de dados constituirá um recurso importante para prossecução de trabalho de investigação e desenvolvimento nesta área.

O descodificador *single-stream*, áudio e vídeo, e o descodificador *multi-stream* síncrono desenvolvidos, baseiam-se em unidades elementares de fala, definidas ao nível da palavra e modeladas sem contexto através dos modelos de Markov não observáveis do tipo semi-contínuo (SCHMM). Estes modelos são treinados segundo o critério da máxima verosimilhança. O reconhecimento baseia-se no algoritmo de descodificação de Viterbi e permite utilizar gramáticas.

A principal conclusão resultante do trabalho é que num reconhecedor automático de fala a utilização simultânea de múltiplas características, acústicas e visuais pode reduzir, neste caso em particular, assim como em outras aplicações similares, significativamente o valor esperado da taxa de erro (WER). Esta utilização simultânea pode tornar o sistema muito mais eficiente em condições adversas (SNR baixo). Esta eficiência traduz-se numa redução no valor da taxa de erro (WER) do *sistema multi-stream* audio-visual quando comparado com a taxa de erro do sistema *single-stream* áudio.

No desenvolvimento do sistema sentiu-se a necessidade de criar um modelo para eventos extra-linguísticos inerentes à fala do orador, produzidos pelo mesmo durante a gravação da base de dados. Com a inclusão deste modelo verificou-se um ganho de $40\%_{rel}$ no descodificador *single-stream* áudio, sem a utilização de qualquer gramática. O ganho resultou da eliminação de palavras inseridas (a taxa de inserções, I, desceu 35 $\%_{rel}$) pelo descodificador.

A avaliação das gramáticas foi realizada sobre os sistema *single-stream*. Para o *stream* acústico, nas melhores condições de SNR (condições de gravação – 24bB), foi obtido um ganho de $25\%_{rel}$ com a utilização da *Word Pair*. A utilização da *Bigram* levou à redução da perplexidade para 6, tendo-se verificado um ganho de $40\%_{rel.}$. Para o *stream* visual os ganhos obtidos foram de $15\%_{rel}$ com a utilização da *Word Pair* e $25\%_{rel}$ com a utilização da *Bigram*. Estes valores justificam-se eventualmente pelo facto da taxa de erro no *stream* visual ser bastante superior à do *stream* áudio, uma vez que, com a consulta da gramática, uma palavra errada (que ocorre muito mais no *stream* visual) pode originar erros nas palavras que lhe sucedem.

A avaliação individual dos sistemas *single-stream* audio e vídeo foi fundamental para a escolha da abordagem *multi-stream* a desenvolver. O sistema audio apresentou um desempenho final ao nível dos sistemas com tarefas de reconhecimento e especificações semelhantes. Os valores

da WER, obtidos com a *Bigram*, foram de 1,12%$_{abs}$ no conjunto de desenvolvimento e de 1,84%$_{abs}$ no conjunto de validação. Estes valores são aceitáveis dadas as condicionantes do orador (dada a sua doença a relação SNR obtida de aproximadamente 25dB, é um valor relativamente baixo para as condições de gravação) e da própria base de dados (a sua dimensão reduzida não permitiu realizar uma segmentação óptima de modo a ter certezas da avaliação correcta de todos os modelos). O sistema vídeo apresentou uma WER de 36,88%$_{abs}$ no conjunto de desenvolvimento e de 38,44%$_{abs}$ no conjunto de validação. É difícil avaliar correctamente este desempenho uma vez que os trabalhos realizados no reconhecimento de fala visual são reduzidos (quando comparados com os sistemas acústicos). O desempenho dos sistemas semelhantes ao desenvolvido neste trabalho apresentam uma WER com valores entre os 30%$_{abs}$ e 40%$_{abs}$ [57][129][157], apresentando uma taxa menor com técnicas de pós-processamento visual que neste trabalho não foram implementadas. Dados os resultados estarem ao nível do estado da arte, para sistemas com especificações semelhantes, apesar da WER elevada, a inclusão da componente visual da fala melhorou o desempenho do sistema, como seria de esperar.

A importância da inclusão da componente visual em condições acústicas desfavoráveis (baixo SNR) ficou evidente através do ganho obtido pelo sistema multi-stream, tendo-se verificado um ganho de aproximadamente 30%$_{rel}$ para valores de SNR de 14dB e 9dB. Verificou-se que quanto menor for a SNR, maior é o benefício da inclusão da componente visual, tendo-se atingido um ganho de 45%$_{rel}$ para um SNR de 4dB. Estes valores foram determinados em função do conjunto de validação, sobre as frases que o sistema não conhecia, tendo sido utilizada a gramática *Bigram*.

Na implementação *multi-stream* realizada verificou-se que a ponderação (peso relativo) de cada *stream* no descodificador depende da SNR acústica, confirmando-se a expectativa da maior importância do *stream* visual quanto menor for a SNR. Verificou-se no entanto que essa ponderação, em situações de SNR habituais[1], tenderá para o *stream* acústico, uma vez que é o *stream* com melhor desempenho[2].

Por outro lado, provou-se também o benefício da utilização das características visuais em ambientes limpos, uma vez que palavras que no *stream* áudio apresentavam confusão, no *stream* visual eram facilmente diferenciáveis. Desse modo, e com uma correcta ponderação dos pesos de cada *stream*, foi possível obter um ganho 20%$_{rel}$ nas melhores condições de SNR,

[1] Depende do ambiente final em que a aplicação vai funcionar

[2] É importante referir que a WER do sistema visual é superior a 30%$_{abs}$, não sendo viável a implementação de um sistema automático de reconhecimento da fala com um desempenho tão baixo, ou seja, considerar a componente visual a principal

quando o descodificador *single-stream*[1] já apresentava uma WER de 1,12%$_{abs}$ no conjunto de desenvolvimento e 1,84%$_{abs}$ no conjunto de validação.

Com este trabalho ficou provado que a combinação das características acústicas e visuais num reconhecedor automático de fala proporciona um melhor desempenho quando comparado com um reconhecedor de fala *standard*, que se traduziu em 0.40%$_{abs}$, 0.7%$_{abs}$, 4.88%$_{abs}$, 7,56%$_{abs}$ e 15.80%$_{abs}$ para 24dB, 19dB, 14dB, 9dB e 4dB, respectivamente.

5.2. Trabalho futuro

Analisando o sistema desenvolvido e os resultados obtidos, é possível levantar um conjunto de aspectos que podem ser explorados de modo a investigar a sua influência no desempenho do sistema.

Esses aspectos podem ser de duas naturezas, ou ao nível da extracção de características, ou ao nível do módulo classificação (integração audio-visual).

No que diz respeito à extracção de características visuais, dois aspectos devem ser referidos. A primeira abordagem seguida foi de baixo nível, com uma técnica baseada numa função de transformação sobre os pixeis da ROI. Da inserção de apenas dois parâmetros[2], altura e largura da boca, juntamente com as características calculadas na primeira abordagem resultou uma pequena melhoria no desempenho do classificador. Esta observação, leva a pensar que a exploração de novas características visuais, e a sua combinação com os parâmetros retirados pela aplicação da DCT, poderá ter um impacto ainda mais favorável. Assim sendo, poderá ser utilizada outra abordagem para extracção das características visuais, como por exemplo a extracção de um modelo paramétrico como referido na secção 2.5.2. O segundo aspecto prende-se com o facto do algoritmo de determinação da ROI ser lento. Este aspecto deve ser levado em consideração se se pretender uma aplicação do sistema a funcionar em tempo real.

Em relação ao módulo de classificação vários aspectos podem ser investigados. No desenvolvimento dos modelos visuais, um estudo a curto prazo seria o de analisar, tal como foi feito no caso acústico, o número óptimo de estados por palavra. Durante o desenvolvimento do sistema visual foi assumido um número igual de estados para o modelo acústico e visual, de modo a simplificar a abordagem inicial de implementação do sistema *multi-stream*. Como é óbvio, a variabilidade associada aos sinais, acústico e visual, de cada palavra não é a mesma.

[1] Na situação com gramática *Bigram*
[2] Modelo híbrido

Como o número de estados depende dessa variabilidade, esse valor deverá ser diferente para as duas abordagens, justificando assim uma exploração deste ponto a curto prazo.

A nível das características acústicas poderia ser explorada como unidade básica de fala, o fonema, em vez da abordagem à palavra implementada. A abordagem utilizando como unidade básica de fala o fonema apresenta normalmente melhores resultados em relação aos sistemas implementados à palavra. No entanto, uma vez que o vocabulário deste sistema é de pequena dimensão e como a base de dados foi projectada para uma abordagem à palavra, desconhece-se se esta possui dados suficientes para uma correcta abordagem ao fonema (teria que ser realizado um estudo prévio para aferir essa possibilidade e, caso necessário, proceder à gravação de novos dados). Não se deve esquecer que, dados os objectivos específicos deste sistema, teve que ser criada, segmentada e etiquetada uma base de dados. Seguindo uma abordagem ao fonema a segmentação seria um processo bastante mais moroso.

Ainda a nível da integração audio-visual, um dos pontos a investigar é a variação do desempenho do classificador, dependendo do grau de liberdade relaxando a imposição de sincronismo nas fronteiras das unidades linguísticas básicas (recorde-se que o sistema obriga a um sincronismo na zona de transição entre essas unidades).

Um ponto interessante a explorar será uma abordagem baseada em técnicas mais sofisticadas de *Decision fusion*. Nestas técnicas a integração pode ser realizada ao nível da classificação, após se obterem os resultados da classificação individual dos dois classificadores *single-stream* do tipo *N-best* sendo utilizado um classificador final para integrar os resultados de cada *stream*. O trabalho realizado permite a integração rápida desta abordagem uma vez que foram desenvolvidos dois sistemas *single-stream* (seria necessário alterar o algoritmo *single-stream* de modo a permitir que a saída do classificador seja um ranking de hipóteses). É de esperar que o desenvolvimento de um classificador final permita retirar conclusões importantes acerca da combinação entre o *stream* áudio e vídeo. A longo prazo deverá ser testado o sistema com algoritmos de descodificação mais complexos, como por exemplo, o *two-level DP*.

A nível da modelação audio-visual poderá ser encetado um novo caminho, baseado em *Dynamic Bayesian Networks* (DBNs), que conduzirão a uma melhor modelação conjunta dos *streams*. Esta abordagem tem apresentado resultados promissores no reconhecimento audio-visual de fala.

Na apresentação da abordagem *multi-stream* foi referido o aumento da complexidade do sistema, que leva a um aumento consequente do tempo de processamento. Verificou-se um aumento computacional relativo três vezes superior na abordagem *multi-stream* (com os *streams* audio e vídeo), em relação a abordagem *single-stream*. Uma análise pormenorizada

das rotinas onde o sistema gasta mais tempo poderá permitir um aperfeiçoamento do código, assim como a implementação de hardware específico[1], que permitam reduzir o tempo de processamento.

Por último, uma análise interessante a efectuar a longo prazo será a da avaliação do sistema com ruído visual, ou com imagens apresentando outras características[2]. A avaliação do sistema visual sob diferentes condições será importante para robustecer o sistema quando este não funcionar nas condições ideais (de desenvolvimento). Para além do ruído visual, outros aspectos a analisar serão a avaliação com iluminações diferentes, ângulos diferentes de aquisição da ROI e detecção da ROI com planos de fundo diferentes, entre outras.

[1] DSPs, FPGAs, entre outras

[2] Imagens adquiridas com outras câmaras de vídeo e resoluções diferentes (por exemplo, webcam.)

ANEXO A

Constituição das frases da base de dados LPFAV2

De seguida é apresentada a constituição de cada frase da base de dados LPFAV2 gravada em função das palavras do seu vocabulário.

24112003_01_01: dezanove mil quatrocentos e setenta e quatro virgula novecentos e quarenta e um mais cinco elevado ao quadrado igual a
24112003_01_02: um milhao oitocentos e dezasseis mil a dividir por logaritmo base dez de trezentos e quarenta e oito igual a
24112003_01_03: zero virgula zero zero um apaga tudo um elevado a menos dois vezes dois milhoes igual a
24112003_01_04: onze mil setecentos e vinte cinco virgula cento e cinquenta e nove menos logaritmo base seis cinquenta e tres igual a
24112003_01_05: dezasseis mil e cem virgula seiscentos e dezassete a dividir por quinze milhoes apaga ultima noventa e cinco igual a
24112003_01_06: cem mil quinhentos e dezanove virgula duzentos oitenta e oito mais raiz quadrada mil oitocentos e setenta igual a
24112003_01_07: vinte e dois milhoes setecentos e dezanove virgula cem vezes sete elevado ao quadrado apaga tudo
24112003_01_08: cem mil novecentos e quinze virgula cento e dez vezes raiz cubica de setecentos e oitenta e nove igual a
24112003_01_09: mil setecentos e dezanove virgula trezentos e treze a dividir por logaritmo de cem igual a
24112003_01_10: um milhao oitocentos e quinze mais abrir parenteses inverso trinta menos inverso seis ao quadrado fechar parenteses igual a
24112003_01_11: cinco milhoes virgula cento e catorze apaga tudo dois ao quadrado vezes tres ao cubo igual a
24112003_01_12: oito virgula seiscentos e dezoito vezes apaga ultima mais inverso sessenta e oito igual a
24112003_01_13: dezasseis virgula novecentos e dezassete menos dez elevado sete igual a
24112003_01_14: um milhao novecentos e treze mil vezes abrir parenteses logaritmo base dez de oitocentos mais dois indice quatro de quatrocentos fechar parenteses igual a
24112003_01_15: por um milhao novecentos e treze mil vezes abrir parenteses logaritmo base dez de oitocentos mais raiz indice quatro de quatrocentos fechar parenteses igual a
24112003_01_16: mil quatrocentos e dez elevado a cinquenta e seis mais logaritmo base dez inverso sete apaga ultima logaritmo de dezassete milhoes igual a
24112003_01_17: um milhao novecentos e treze mil vezes abrir parenteses logaritmo base oitocentos mais raiz indice quatro de quatrocentos fechar parenteses igual a
24112003_01_18: mil quatrocentos e dez elevado a cinquenta seis mais logaritmo base inverso sete apaga ultima logaritmo de dezassete milhoes igual a
24112003_01_19: raiz quadrada mil duzentos e quinze menos raiz indice quatro de quatrocentos e vinte seis igual a
24112003_01_20: raiz cubica mil quinhentos e setenta e nove a dividir por oitocentos e setenta elevado ao cubo menos inverso cem igual a
24112003_01_21: logaritmo mil trezentos e treze elevado a quinze vezes raiz quadrada zero virgula oito apaga tudo
24112003_01_22: um virgula duzentos e dezassete mais logaritmo base dez inverso oito apaga ultima logaritmo cinco ao cubo igual a
24112003_01_23: um virgula novecentos e treze elevado oito menos raiz indice sete de logaritmo de setenta e oito ao cubo apaga ultima elevado a nove igual a
24112003_01_24: mil setecentos e dezassete a dividir por logaritmo base dez quinhentos e noventa e oito menos sessenta e nove igual a
24112003_01_25: quinhentos e catorze mil duzentos e dezanove virgula novecentos e onze vezes logaritmo de quinze elevado a sete igual a
24112003_02_01: quinhentos e catorze mil duzentos e dezanove virgula novecentos e onze vezes logaritmo de quinze elevado a sete igual a
24112003_02_02: um milhao seiscentos e dezoito mil quatrocentos e quinze virgula setecentos e catorze a dividir por cem apaga ultima nove ao cubo igual a
24112003_02_03: um milhao seiscentos e dezoito mil quatrocentos e quinze virgula setecentos e catorze a dividir por cem apaga ultima nove ao cubo igual a
24112003_02_04: trezentos e doze mil seiscentos e dezasseis virgula oitocentos e doze mais abrir parenteses raiz indice vinte de setecentos fechar parenteses igual a
24112003_02_05: cento e dezassete mil quinhentos e dezanove virgula setecentos e onze a dividir por logaritmo base dez de um milhao igual a
24112003_02_06: dezassete milhoes quatrocentos e onze mil setecentos e quinze virgula seiscentos e dezanove menos treze ao quadrado igual a
24112003_02_07: oitocentos e doze mil quatrocentos e doze

24112003_02_08: dois milhoes setecentos e catorze mil quatrocentos e treze virgula quinhentos e quinze a dividir por raiz indice sete de quarenta e cinco igual a
24112003_02_09: um milhao trezentos e dezoito mil duzentos e treze vezes zero virgula quinhentos e cinquenta e tres apaga tudo
24112003_02_10: oito milhoes novecentos e quinze mil quatrocentos e dezassete virgula trezentos e onze vezes raiz cubica de cinco elevado a doze igual a
24112003_02_11: trezentos e dez mil quinhentos e onze virgula trezentos e treze a dividir por apaga ultima mais logaritmo base dez de dezanove milhoes igual a
24112003_02_12: inverso setecentos e catorze mil seiscentos e quinze mais inverso de dez elevado dois menos vinte e um igual a
24112003_02_13: vinte e sete milhoes seiscentos e catorze mil setecentos e dez virgula quinhentos e dezasseis a dividir por raiz indice oito de setecentos e sessenta e nove igual a
24112003_02_14: quatrocentos e dezanove mil trezentos e dezassete virgula duzentos e quinze menos abrir parenteses quinhentos e vinte ao quadrado mais zero virgula seis elevado a cinco fechar parenteses igual a
24112003_02_15: quatrocentos e quinze mil seiscentos e onze virgula zero treze vezes logaritmo de noventa e sete mais treze ao quadrado apaga ultima ao cubo igual a
24112003_02_16: um milhao oitocentos e catorze mil e cem virgula duzentos e dezanove a dividir por raiz vinte e sete de zero virgula dezoito elevado doze dezasseis igual a
24112003_02_17: quinhentos e treze mil duzentos e quinze virgula trezentos e catorze vezes dez apaga ultima logaritmo base dez de dezasseis ao quadrado igual a
24112003_02_18: tres milhoes setecentos e dezasseis mil novecentos e doze virgula cento e dez vezes inverso cem menos logaritmo de quarenta e sete igual a
24112003_02_19: seiscentos e dezassete mil setecentos e dezanove virgula quinhentos e catorze a dividir por dois ao cubo mais noventa milhoes elevado a cem apaga ultima igual a
24112003_02_20: cinquenta milhoes duzentos e doze mil quinhentos e catorze virgula seiscentos e dezassete mais raiz cubica de quinhentos igual a
24112003_02_21: trezentos e dez mil cento e treze virgula duzentos e onze a dividir por raiz quadrada de cinquenta e nove virgula zero igual a
24112003_02_22: oitocentos e quinze mil quatrocentos e dezassete virgula quinhentos e doze menos apaga tudo inverso quatro vezes inverso oito elevado a mais sete igual a
24112003_02_23: um milhao novecentos e catorze mil setecentos e doze virgula quatrocentos e dez vezes logaritmo de trinta e nove igual a
24112003_02_24: quinhentos e dez mil seiscentos e quinze virgula setecentos e dezasseis a dividir por oitenta e cinco menos raiz quadrada de dezoito igual a
24112003_02_25: oitenta milhoes setecentos e doze mil cento e dezassete virgula quinhentos e onze vezes logaritmo base dez de setenta milhoes apaga tudo
24112003_03_01: duzentos e onze mil quinhentos e dez virgula novecentos e quinze vezes abrir parenteses raiz indice oito de mil e quinhentos mais raiz quadrada de catorze fechar parenteses igual a
24112003_03_02: doze elevado a quatro apaga ultima sete mais raiz quadrada de dezoito igual a
24112003_03_03: sete vezes vinte apaga tudo onze vezes cem igual a
24112003_03_04: cento e dezoito mil e cem virgula setecentos e treze mais raiz quadrada vinte seis igual a
24112003_03_05: um milhao vezes cento e doze mil apaga ultima quinze elevado a dez igual a
24112003_03_06: treze milhoes novecentos e dezanove mil trezentos e nove virgula zero catorze vezes cento e noventa e dois elevado doze igual a
24112003_03_07: quatrocentos e dezanove mil duzentos e dez virgula zero treze menos raiz indice cinco de vinte sete mil ao cubo igual a
24112003_03_08: cinco milhoes quatrocentos e dezoito mil duzentos e dezanove virgula cem a dividir por sessenta e oito elevado a catorze igual a
24112003_03_09: um milhao duzentos e dezanove mil setecentos e catorze virgula trezentos e um vezes raiz quadrada de cem apaga ultima inverso cem igual a
24112003_03_10: trezentos e doze mil duzentos e onze virgula quatrocentos e quarenta e seis a dividir por abrir parenteses treze ao quadrado vezes de dezassete ao cubo fechar parenteses igual a
24112003_03_11: um milhao duzentos e vinte mil menos abrir parenteses raiz quadrada novecentos e noventa e nove fechar parenteses mais cinco elevado a dez igual a
24112003_03_12: novecentos e dezasseis mil quinhentos e dezanove virgula cem vezes tres milhoes ao cubo apaga ultima ao quadrado igual a
24112003_03_13: trezentos e dezanove mil setecentos e dezoito virgula novecentos e dezassete mais raiz indice sete de cem igual a
24112003_03_14: um milhao seiscentos e dez mil quinhentos e dezassete virgula novecentos e quinze vezes abrir parenteses inverso trinta mais dois elevado a quinze fechar parenteses igual a
24112003_03_15: quatrocentos e dezasseis mil duzentos e treze a dividir por raiz indice sessenta e cinco de trezentos e oitenta e nove ao cubo igual a
24112003_03_16: um ao quadrado apaga tudo nove ao quadrado menos dezasseis elevado a doze igual a
24112003_03_17: cinco milhoes setecentos e dezasseis mil oitocentos e dezoito virgula novecentos e dezanove menos apaga ultima a dividir por raiz quadrada de trinta e sete elevado oito igual a
24112003_03_18: um milhao seiscentos e onze mil trezentos e doze virgula oitocentos e dez mais quarenta e sete mil e cem vezes seiscentos e sete mil elevado a cinquenta igual a
24112003_03_19: cinco milhoes setecentos e dezasseis mil oitocentos e dezoito virgula novecentos e dezanove menos apaga ultima a dividir por raiz quadrada de trinta e sete elevado a oito igual a
24112003_03_20: um milhao seiscentos e onze mil trezentos e doze virgula oitocentos e dez mais quarenta e sete mil e cem vezes seiscentos e sete mil elevado a cinquenta igual a
24112003_03_21: trezentos e dezoito mil cento e quinze virgula quatrocentos e doze vezes raiz indice noventa de dezassete ao quadrado igual a
24112003_03_22: quatro milhoes seiscentos e doze mil oitocentos e dezanove dividir por oitenta e nove mais raiz cubica de dez igual a
24112003_03_23: quinze virgula oito mais abrir parenteses inverso novecentos e vinte menos cinquenta e sete elevado dezasseis fechar parenteses igual a
24112003_03_24: nove milhoes quatrocentos e dezanove mil setecentos e catorze virgula seiscentos e dezasseis mais dezoito elevado a cinco igual a
24112003_03_25: trezentos e catorze mil quatrocentos e treze virgula setecentos e dezanove vezes um milhao e oitocentos mil apaga ultima duzentos mil igual a
24112003_04_01: trinta elevado a quatro mais abrir parenteses raiz indice oito de dezassete a dividir por raiz cubica de vinte e dois menos catorze ao cubo fechar parenteses igual a
24112003_04_02: novecentos e onze mil seiscentos e dezoito virgula novecentos e nove a dividir por cinco mais inverso de dezanove apaga ultima igual a
24112003_04_03: um milhao oitocentos e treze mil seiscentos e quinze menos raiz indice cinquenta de de setecentos e vinte mil apaga tudo
24112003_04_04: dezanove mil quatrocentos e setenta e quatro virgula novecentos e quarenta e um mais cinco elevado ao quadrado igual a
24112003_04_05: um milhao oitocentos e dezasseis mil a dividir por logaritmo base dez de trezentos e quarenta e oito igual a
24112003_04_06: zero virgula zero zero um apaga tudo um elevado a menos dois vezes dois milhoes igual a
24112003_04_07: onze mil setecentos e vinte e cinco virgula cento e cinquenta e nove menos logaritmo base seis cinquenta e tres igual a
24112003_04_08: dezassete mil e cem virgula seiscentos e dezassete a dividir por quinze milhoes apaga ultima noventa e cinco igual a
24112003_04_09: cem mil quinhentos e dezanove virgula duzentos e oitenta e oito mais raiz quadrada de mil oitocentos e setenta de igual a
24112003_04_10: vinte e dois milhoes setecentos e dezanove virgula cem vezes sete elevado ao quadrado apaga tudo
24112003_04_11: cem mil novecentos e quinze virgula cento e dez vezes raiz cubica de setecentos oitenta e nove igual a
24112003_04_12: mil setecentos e dezanove virgula trezentos e treze a dividir por logaritmo de cem igual a
24112003_04_13: um milhao oitocentos e quinze mais abrir parenteses inverso trinta menos inverso de dez ao quadrado fechar parenteses igual a
24112003_04_14: cinco milhoes virgula cento e catorze apaga tudo dois ao quadrada vezes tres ao cubo igual a
24112003_04_15: oito virgula seiscentos e dezoito vezes apaga ultima mais inverso sessenta e oito igual a
24112003_04_16: dezasseis virgula novecentos e dezassete menos dez elevado a sete igual a
24112003_04_17: um milhao novecentos e treze mil vezes abrir parenteses logaritmo base dez oitocentos mais raiz indice quatro de quatrocentos fechar parenteses igual a
24112003_04_18: a mil quatrocentos e dez elevado a cinquenta e seis mais logaritmo base dez de inverso sete apaga ultima logaritmo de dezassete milhoes igual a
24112003_04_19: raiz quadrada de mil e duzentos e quinze menos raiz indice quatro de quatrocentos e vinte seis igual a
24112003_04_20: raiz cubica de mil quinhentos e setenta e nove a dividir por oitocentos e sessenta elevado ao cubo menos inverso de um milhao igual a
24112003_04_21: logaritmo de mil trezentos e treze elevado a quinze vezes raiz quadrada de zero virgula oito apaga tudo
24112003_04_22: um virgula novecentos e treze elevado a quinhentos menos raiz indice sete de logaritmo de setenta e oito ao cubo apaga ultima elevado dezanove igual a
24112003_04_23: um virgula duzentos e dezassete mais logaritmo base dez de inverso oito apaga ultima logaritmo de cinco ao cubo igual a
24112003_04_24: mil setecentos e dezassete a dividir por logaritmo base dez de quinhentos e noventa e oito menos sessenta e nove igual a
24112003_04_25: um doze elevado a quatro apaga ultima sete mais raiz quadrada dezoito igual a
24112003_05_01: sete vezes vinte apaga tudo onze vezes cem igual a
24112003_05_02: cento e dezoito mil e cem virgula setecentos e treze mais raiz quadrada vinte seis igual a

24112003_05_03: um milhao vezes cento e doze mil apaga ultima quinze elevado dez igual a
24112003_05_04: tres milhoes novecentos e dezanove mil trezentos e nove virgula zero catorze vezes cento e noventa e dois elevado a doze igual a
24112003_05_05: quatrocentos e dezanove mil duzentos e dez virgula zero treze menos raiz indice cinco de vinte sete mil ao cubo igual a
24112003_05_06: cinco milhoes quatrocentos e dezoito mil duzentos e dezanove virgula cem a dividir por cento e oito elevado a catorze igual a
24112003_05_07: um milhao duzentos e dezanove mil setecentos e catorze virgula trezentos e um vezes raiz quadrada cem apaga ultima inverso cem igual a
24112003_05_08: trezentos e doze mil duzentos e onze virgula quatrocentos e quarenta e seis a dividir por abrir parenteses tres ao quadrado vezes dezassete ao cubo fechar parenteses igual a
24112003_05_09: um milhao duzentos e vinte mil menos abrir parenteses raiz quadrada de novecentos e noventa e nove fechar parenteses mais cinco elevado a dez igual a
24112003_05_10: novecentos e dezasseis mil quinhentos e dezanove virgula cem vezes tres milhoes ao cubo apaga ultima ao quadrado igual a
24112003_05_11: trezentos e dezanove mil setecentos e dezoito virgula novecentos e dezasseis
24112003_05_12: um milhao seiscentos e dez mil quinhentos e dezassete virgula novecentos e quinze vezes abrir parenteses inverso de trinta mais dois elevado a quinze fechar parenteses igual a
24112003_05_13: quatrocentos e dezasseis mil duzentos e treze a dividir por raiz indice sessenta e cinco de trezentos e oitenta e nove ao cubo igual a
24112003_05_14: um ao quadrado apaga tudo nove ao quadrado menos de dezasseis elevado a doze igual a
24112003_05_15: cinco milhoes setecentos e dezasseis mil oitocentos e dezoito virgula novecentos e dezanove menos apaga ultima a dividir por raiz quadrada de trinta e sete elevado oito igual a
24112003_05_16: um milhao seiscentos e onze mil trezentos e doze virgula oitocentos e dez mais quarenta e sete mil e cem vezes seiscentos e sete mil elevado a cinquenta igual a
24112003_05_17: trezentos e dezoito mil cento e quinze virgula quatrocentos e doze vezes raiz indice noventa de dezassete ao quadrado igual a
24112003_05_18: quatro milhoes seiscentos e doze mil oitocentos e dezanove a dividir por oitenta e nove mais raiz cubica dez igual a
24112003_05_19: quinze virgula oito mais abrir parenteses inverso de novecentos e vinte menos cinquenta e sete elevado a seis fechar parenteses igual a
24112003_05_20: nove milhoes quatrocentos e dezanove mil setecentos e catorze virgula seiscentos e dezasseis mais dezoito elevado a cinco igual a
24112003_05_21: trezentos e catorze mil quatrocentos e treze virgula setecentos e dezanove vezes um milhao e oitocentos mil apaga ultima duzentos mil igual a
24112003_05_22: trinta elevado a quatro mais abrir parenteses raiz indice oito de dezassete a dividir por raiz cubica de vinte e dois menos catorze ao cubo fechar parenteses igual a
24112003_05_23: novecentos e onze de mil seiscentos e dezoito virgula novecentos e nove e dividir por cinco cubo mais inverso de dezanove apaga ultima igual a
24112003_05_24: novecentos e onze mil seiscentos e dezoito virgula novecentos e nove a dividir por cinco cubo mais inverso de dezanove apaga ultima igual a
24112003_05_25: um milhao oitocentos e treze mil seiscentos e quinze menos raiz indice cinquenta de setecentos e vinte mil apaga tudo
22122003_06_01: novecentos e doze mil cento e quinze a dividir por abrir parenteses cinco elevado a sete fechar parenteses igual a
22122003_06_02: cento e dezanove mil trezentos e onze virgula seiscentos e dez mais setenta elevado a vinte e dois igual a
22122003_06_03: cem mil quinhentos e dezanove virgula cem a dividir por apaga tudo
22122003_06_04: raiz cubica de setecentos milhoes igual a
22122003_06_05: trezentos e treze mil novecentos e catorze virgula quinhentos e sessenta e tres menos oito ao quadrado igual a
22122003_06_06: cem mil duzentos e dezasseis virgula oitocentos e doze vezes apaga ultima mais raiz indice cinco de noventa e oito igual a
22122003_06_07: quatrocentos e doze mil trezentos e dezasseis virgula seiscentos e quarenta e seis a dividir por doze ao cubo apaga tudo
22122003_06_08: cem mil quatrocentos e dezoito virgula setecentos e dezanove vezes abrir parenteses inverso cem mais inverso de dois milhoes igual a
22122003_06_09: cento e dezasseis mil seiscentos e quinze virgula cem vezes apaga tudo raiz quadrada oito a dividir por raiz cubica nove igual a
22122003_06_10: oitocentos e noventa e nove mil trezentos e dezassete virgula cento e dezoito a dividir por zero ao quadrado igual a
22122003_06_11: duzentos e dezoito apaga tudo mil oitocentos e dez virgula setecentos e onze mais inverso quinze elevado a quarenta igual a
22122003_06_12: cem mil novecentos e dezanove virgula setecentos e catorze a dividir por dezassete milhoes apaga tudo
22122003_06_13: quatrocentos e dezasseis mil quinhentos e treze virgula duzentos e noventa e cinco menos dois ao quadrado igual a
22122003_06_14: trezentos e dezassete mil quatrocentos e dezoito virgula cem vezes inverso de oitenta ao cubo igual a
22122003_06_15: cem mil seiscentos e treze virgula setecentos e vinte e nove a dividir por cinquenta e quatro elevado zero igual a
22122003_06_16: quinhentos e quinze mil novecentos e dezoito virgula cento e doze vezes inverso tres milhoes apaga ultima inverso catorze milhoes igual a
22122003_06_17: quatrocentos e dezassete mil trezentos e quinze vezes abrir parenteses nove ao quadrado menos raiz indice seis de cem milhoes fechar parenteses igual a
22122003_06_18: setecentos e dez mil seiscentos e dezassete virgula novecentos e onze a dividir por sete ao cubo igual a
22122003_06_19: trezentos e dezoito mil duzentos e dez virgula zero doze mais sete ao quadrado igual a
22122003_06_20: cento e dezasseis mil oitocentos e dezanove a dividir por abrir parenteses sete ao cubo mais oito ao quadrado fechar parenteses igual a
22122003_06_21: seiscentos e dez mil setecentos e quinze virgula oitocentos e dezasseis menos catorze elevado a dez igual a
22122003_06_22: cem mil cento e dezanove virgula duzentos e noventa e sete vezes dezasseis elevado a nove apaga tudo
22122003_06_23: novecentos e onze mil quinhentos e dezoito virgula oitocentos e dezassete a dividir por raiz cubica de um milhao igual a
22122003_06_24: cem mil seiscentos e dezassete virgula setecentos e treze vezes raiz indice quatro de oitocentos e noventa igual a
22122003_06_25: duzentos e quinze mil quatrocentos e onze virgula novecentos e doze vezes inverso cem igual a
22122003_07_01: quatrocentos e catorze mil trezentos e dezoito virgula cento e dezasseis a dividir por dois ao quadrado apaga ultima ao cubo igual a
22122003_07_02: quatrocentos e onze mil seiscentos e quinze virgula duzentos e dezoito mais tres milhoes elevado a dezanove igual a
22122003_07_03: novecentos e doze mil quatrocentos e dez virgula setecentos e dezanove a dividir por raiz cubica de duzentos igual a
22122003_07_04: cento e dezasseis mil novecentos e onze virgula seiscentos e doze menos apaga ultima mais inverso cinco milhoes igual a
22122003_07_05: duzentos e dezanove mil oitocentos e dez virgula trezentos e catorze vezes dois ao cubo igual a
22122003_07_06: seiscentos e dezoito mil quatrocentos e onze virgula quinhentos e dezoito a dividir por cinco elevado oito igual a
22122003_07_07: cento e treze mil quatrocentos e dezanove virgula nove cento e dez vezes quinze milhoes ao quadrado apaga tudo
22122003_07_08: quinhentos e vinte sete mil seiscentos e treze virgula cem vezes inverso de oitenta e nove igual a
22122003_07_09: oitocentos e dezoito mil duzentos e quinze virgula cento e dezasseis a dividir por sete elevado a onze igual a
22122003_07_10: cento e dez mil quatrocentos e dezassete virgula seiscentos e onze mais catorze elevado a dezoito igual a
22122003_07_11: setecentos e onze mil cento e doze virgula duzentos e dezanove a dividir por raiz quadrada setenta milhoes igual a
22122003_07_12: cento e dez mil e cem virgula oitocentos e doze menos abrir parenteses cinco ao cubo vezes treze ao quadrado fechar parenteses igual a
22122003_07_13: trezentos e dezanove mil quatrocentos e dezoito virgula setecentos e treze vezes inverso de cem igual a
22122003_07_14: duzentos e doze mil quinhentos e dezanove virgula seiscentos e onze a dividir por quinhentos milhoes apaga tudo
22122003_07_15: abrir parenteses mil novecentos e dezasseis menos inverso cem fechar parenteses a dividir por dois ao cubo igual a
22122003_07_16: inverso mil duzentos e dezassete virgula quatrocentos e catorze apaga ultima tres milhoes e setecentos igual a
22122003_07_17: mil cento e treze apaga ultima um milhao e e trinta e tres mais inverso de dezasseis igual a
22122003_07_18: mil setecentos e dezasseis virgula cento e tres mais nove ao quadrado igual a
22122003_07_19: raiz indice seis de mil trezentos e treze mais dezoito elevado a setenta igual a
22122003_07_20: mil setecentos e dezasseis virgula cento e dezassete vezes dois ao quadrado igual a
22122003_07_21: abrir parenteses mil quinhentos e dezoito a dividir por zero virgula seis fechar parenteses menos cinquenta e nove ao cubo igual a
22122003_07_22: mil setecentos e catorze mais abrir parenteses raiz quadrada de nove menos raiz cubica de oito fechar parenteses elevado a cinco igual a
22122003_07_23: abrir parenteses mil cento e dezasseis vezes inverso sete milhoes fechar parenteses a dividir por dois ao quadrado igual a
22122003_07_24: raiz indice nove de mil oitocentos e quinze virgula quinhentos e catorze menos raiz quadrado doze quatro igual a
22122003_07_25: inverso de sessenta sessenta e cinco mil duzentos e quarenta e seis virgula oitocentos e vinte e tres apaga tudo
22122003_08_01: inverso de sessenta e cinco mil duzentos e quarenta e seis virgula oitocentos e vinte e tres apaga tudo
22122003_08_02: abrir parenteses raiz cubica de noventa e oito mil e cem virgula trezentos e dez fechar parenteses ao cubo igual a
22122003_08_03: raiz cubica de oitenta e quatro mil quatrocentos e dez virgula seiscentos e dezanove elevado oito igual a
22122003_08_04: inverso de quarenta e tres mil quinhentos e sessenta e nove virgula trezentos e um apaga tudo

22122003_08_05: raiz quadrada de oitenta e seis mil trezentos e dezanove virgula seiscentos e nove igual a
22122003_08_06: raiz quadrada de sessenta e dois mil quinhentos e setenta e dois virgula quatrocentos e quinze igual a
22122003_08_07: cento e doze mil duzentos e trinta e sete virgula duzentos e setenta e um elevado a trinta e quatro igual a
22122003_08_08: sessenta e tres mil setecentos e dezassete virgula novecentos e setenta e oito apaga ultima setenta e nove milhoes mais inverso zero virgula por zero um igual a
22122003_08_09: raiz indice seis de vinte e quatro mil trezentos e cinquenta e dois virgula quatrocentos e trinta e um apaga tudo
22122003_08_10: inverso de doze mil e cem virgula trezentos e quarenta e dois vezes raiz quadrada de oitenta e nove igual a
22122003_08_11: raiz indice sete de cem mil quatrocentos e dezanove virgula seiscentos e cinquenta e quatro igual a
22122003_08_12: mil seiscentos e trinta e cinco virgula cento e sessenta e dois e dividir por raiz cubica de cinquenta e cinco apaga tudo
22122003_08_13: dezanove mil quatrocentos e setenta e quatro virgula novecentos e quarenta e um mais cinco elevado ao quadrado igual a
22122003_08_14: um milhao oitocentos e dezasseis mil a dividir por logaritmo base dez de trezentos e quarenta e oito igual a
22122003_08_15: zero virgula zero zero um apaga tudo um elevado a menos dois vezes dois milhoes igual a
22122003_08_16: onze mil setecentos e vinte cinco virgula cento e cinquenta e nove menos logaritmo indice seis de cinquenta e tres igual a
22122003_08_17: um dezassete mil e cem virgula seiscentos e dezassete a dividir por quinze apaga ultima noventa e cinco igual a
22122003_08_18: cem mil quinhentos e dezanove virgula duzentos e oitenta e oito mais raiz quadrada de mil oitocentos e setenta igual a
22122003_08_19: vinte e dois milhoes mil e cem virgula setecentos e dezanove vezes sete elevado a quatro apaga tudo
22122003_08_20: vinte e dois milhoes mil e cem virgula setecentos e dezanove vezes sete elevado a quatro apaga ultima
22122003_08_21: cem mil novecentos e quinze virgula cento e dez vezes raiz cubica de setecentos e oitenta e nove igual a
22122003_08_22: mil setecentos e dezanove virgula trezentos e treze a dividir por logaritmo cem igual a
22122003_08_23: um milhao oitocentos e quinze mais abrir parenteses inverso trinta menos inverso de dez ao quadrado fechar parenteses igual a
22122003_08_24: cinco milhoes virgula cento e catorze apaga tudo dois ao quadrado vezes tres ao cubo igual a
22122003_08_25: oito virgula seiscentos e dezoito vezes apaga ultima mais inverso de sete sessenta e oito igual a
22122003_09_01: dezasseis virgula novecentos e dezassete menos dez elevado dezassete igual a
22122003_09_02: um milhao novecentos e treze mil vezes abrir parenteses logaritmo base de oitocentos mais raiz indice quatro de quatrocentos fechar parenteses igual a
22122003_09_03: um milhao quatrocentos e dez a dividir por nove elevado a seis igual a
22122003_09_04: mil duzentos e quinze mais abrir parenteses dois ao cubo menos sete elevado a seis igual a
22122003_09_05: abrir parenteses dois milhoes quinhentos e setenta e nove vezes cinquenta ao cubo fechar parenteses mais raiz quadrada de oito igual a
22122003_09_06: raiz indice seis de quatro milhoes trezentos e treze apaga ultima cinco milhoes igual a
22122003_09_07: abrir parenteses um virgula novecentos e treze mais zero virgula vinte e tres fechar parenteses menos inverso de sete setenta igual a
22122003_09_08: um virgula duzentos e dezassete mais zero virgula cinco elevado a seis igual a
22122003_09_09: inverso de um virgula quinhentos e treze a dividir por zero virgula sete igual a
22122003_09_10: abrir parenteses mil setecentos e dezassete mais cinco milhoes fechar parenteses elevado a dez apaga tudo
22122003_09_11: raiz indice quatro de mil oitocentos e quarenta e nove menos inverso de sessenta e quatro igual a
22122003_09_12: um milhao quinhentos e catorze virgula quinhentos e treze elevado a noventa e tres igual a
22122003_09_13: raiz indice cem de um virgula duzentos e vinte e nove vezes raiz quadrada de noventa e nove igual a
22122003_09_14: um virgula cento e treze mais quinze milhoes quatrocentos e setenta e sete apaga tudo
22122003_09_15: inverso de trezentos e dezasseis mil quinhentos e trinta e dois virgula quatrocentos e vinte e quatro igual a
22122003_09_16: abrir parenteses zero virgula quinhentos e catorze ao cubo menos tres virgula dois elevado a sete fechar parenteses a dividir por cinco igual a
22122003_09_17: um virgula zero treze elevado a vinte e dois vezes inverso cinquenta e nove ao quadrado igual a
22122003_09_18: raiz cubica de um virgula quinhentos e nove a dividir por inverso de quarenta e sete ao cubo igual a
22122003_09_19: raiz indice oito doze de um milhao virgula seiscentos e oitenta e nove elevado a seiscentos e sessenta e apaga ultima igual a
22122003_09_20: abrir parenteses zero virgula trezentos e quinze menos zero virgula cinco fechar parenteses elevado a quatro igual a
22122003_09_21: zero virgula cento e treze a dividir por doze virgula vinte apaga tudo
22122003_09_22: raiz quadrada de zero virgula setecentos e catorze menos cinco milhoes elevado a menos tres igual a
22122003_09_23: um virgula oitocentos e treze mais abrir parenteses raiz cubica nove a dividir por sete ao cubo fechar parenteses vezes quarenta igual a
22122003_09_24: raiz indice seis de novecentos e vinte e quatro mil novecentos e dezoito virgula quatrocentos e oitenta e quatro apaga tudo
22122003_09_25: quinze mil seiscentos e cinquenta e cinco virgula cem apaga ultima zero virgula cinco elevado a sete igual a
22122003_10_01: inverso de setenta e nove mil setecentos e catorze virgula quinhentos e setenta e um ao quadrado igual a
22122003_10_02: cem mil setecentos e quinze virgula duzentos e dezoito vezes apaga ultima menos trinta milhoes ao cubo igual a
22122003_10_03: inverso de treze mil e doze virgula trezentos e noventa e um menos doze milhoes ao quadrado igual a
22122003_10_04: raiz indice oito de cinquenta e nove mil seiscentos e dezasseis virgula oitocentos e setenta e tres apaga tudo
22122003_10_05: raiz quadrada de noventa e oito mil quatrocentos e treze virgula cem elevado a dezasseis igual a
22122003_10_06: raiz indice seis de catorze mil e cem virgula novecentos e setenta e tres a dividir por inverso de trinta e sete igual a
22122003_10_07: quarenta e sete mil oitocentos e quinze virgula quinhentos e quarenta e oito elevado a cinco igual a
22122003_10_08: abrir parenteses dezoito milhoes e cem virgula oitocentos e onze menos quarenta e nove fechar parenteses elevado a sete igual a
22122003_10_09: raiz quadrada de seiscentos e catorze mil quinhentos e trinta e nove virgula trezentos e dezoito apaga ultima
22122003_10_10: raiz cubica de cinquenta e dois mil duzentos e dezoito virgula zero vinte e tres apaga tudo
22122003_10_11: abrir parenteses zero virgula duzentos e onze mais um virgula trezentos e dez fechar parenteses vezes dois ao cubo igual a
22122003_10_12: quatrocentos e treze mil cento e tres virgula trezentos e quinze vezes raiz cubica de dezassete igual a
22122003_10_13: trezentos e doze mil oitocentos e dezoito virgula oitocentos e catorze vezes raiz indice treze de trezentos e sessenta e nove igual a
22122003_10_14: quatrocentos e onze mil novecentos e dez virgula duzentos e dezanove a dividir por quinhentos e vinte e sete apaga ultima cinquenta e seis igual a
22122003_10_15: seiscentos e doze mil setecentos e quinze virgula cento e dezoito mais nove elevado a zero igual a
22122003_10_16: um novecentos e sete e dez mil quatrocentos e dezanove virgula duzentos e doze a dividir por inverso de cento e setenta igual a
22122003_10_17: oitocentos e dezassete mil quatrocentos e dezoito virgula quatrocentos e dez menos inverso setenta igual a
22122003_10_18: setecentos e onze mil novecentos e dezasseis virgula oitocentos e dezassete vezes dezasseis elevado a nove igual a
22122003_10_19: quatrocentos e dezanove mil e cem virgula duzentos oitenta e sete a dividir por trezentos milhoes igual a
22122003_10_20: trezentos e dezanove mil quatrocentos e doze virgula cento e onze vezes raiz indice oitenta de quatrocentos e trinta e um igual a
22122003_10_21: quatrocentos e dezanove mil novecentos e doze virgula oitocentos e quinze vezes inverso dois igual a
22122003_10_22: setecentos e treze mil seiscentos e dezassete virgula zero dezanove a dividir por treze ao cubo igual a
22122003_10_23: cento e dezassete mil quatrocentos e quinze virgula seiscentos e dezoito mais quinze milhoes ao quadrado igual a
22122003_10_24: quinhentos e onze mil oitocentos e doze virgula quinhentos e dezasseis a dividir por raiz cubica de trinta e dois igual a
22122003_10_25: novecentos e doze mil e cem virgula setecentos e noventa e dois menos raiz quadrada de dezasseis igual a
22122003_11_01: cento e dezoito mil quatrocentos e cinquenta e cinco virgula cem vezes cem elevado a quatro igual a
22122003_11_02: duzentos e dez mil trezentos e dezasseis virgula trezentos e onze a dividir por apaga tudo
22122003_11_03: oito milhoes menos dez mil igual a
22122003_11_04: cento e quinze mil quinhentos e dez virgula oitocentos e dezoito vezes raiz indice dezanove de um
22122003_11_05: cento e quinze mil quinhentos e dez virgula oitocentos e dezoito vezes raiz indice dezanove de seiscentos e quarenta e um apaga tudo
22122003_11_06: setecentos e doze mil quatrocentos e onze vezes abrir parenteses nove ao cubo menos inverso de quinze fechar parenteses igual a
22122003_11_07: duzentos e dezasseis mil setecentos e dezassete virgula seiscentos e dezanove a dividir por quarenta e sete elevado oito igual a
22122003_11_08: seiscentos e dezassete mil cento e dez virgula cento e catorze mais raiz quadrada de catorze igual a
22122003_11_09: quinhentos e dezanove mil duzentos e catorze virgula quinhentos e doze a dividir por seis ao cubo igual a
22122003_11_10: novecentos e dezoito mil oitocentos e e onze virgula zero catorze menos treze ao quadrado igual a

22122003_11_11: novecentos e doze mil novecentos e quinze virgula cento e doze vezes raiz quadrada de trezentos e vinte igual a
22122003_11_12: trezentos e dez mil seiscentos e dezanove a dividir por dezassete elevado a seis igual a
22122003_11_13: quatrocentos e doze mil trezentos e dezasseis virgula duzentos e dezassete menos raiz quadrada de trinta e tres igual a
22122003_11_14: cento e dezasseis mil oitocentos e dezoito virgula seiscentos e quinze vezes cinco elevado a cinco apaga tudo
22122003_11_15: setecentos e onze mil cento e dezassete virgula duzentos e dezanove a dividir por inverso sete igual a
22122003_11_16: novecentos e setenta e dois mil duzentos e cinquenta e dois virgula cem mais cinco milhoes igual a
22122003_11_17: trezentos e dez mil duzentos e noventa e tres virgula cem a dividir por inverso treze elevado a seis igual a
22122003_11_18: quinhentos e dezasseis mil duzentos e onze virgula setecentos e dez menos dezoito elevado nove igual a
22122003_11_19: cento e dezassete mil duzentos e dezassete virgula quinhentos e dezoito vezes raiz cubica de vinte e sete igual a
22122003_11_20: setecentos e doze mil cento e treze virgula trezentos e dezasseis a dividir por quinze elevado dezanove igual a
22122003_11_21: quinhentos e catorze mil setecentos e dezasseis virgula novecentos e onze vezes dezasseis ao cubo apaga tudo
22122003_11_22: setecentos e dezoito mil oitocentos e dezassete virgula cento e onze mais inverso de cem igual a
22122003_11_23: duzentos e dezasseis mil trezentos e treze virgula oitocentos e nove a dividir por sete ao cubo apaga ultima ao quadrado igual a
22122003_11_24: novecentos e onze mil cento e dezassete apaga ultima dezanove mais inverso de treze igual a
22122003_11_25: quinhentos e dezassete mil cento e dezasseis virgula duzentos e doze a dividir por seis ao cubo igual a
22122003_12_01: quatrocentos e doze mil setecentos e dezoito virgula setecentos e catorze menos vinte ao quadrado apaga tudo
22122003_12_02: oitocentos e catorze mil duzentos e dezasseis vezes abrir parenteses duzentos e um menos quatrocentos e tres fechar parenteses igual a
22122003_12_03: duzentos e onze mil e cem virgula novecentos e quinze a dividir por oito elevado a dezassete apaga tudo
22122003_12_04: trezentos e doze mil cento e quinze virgula oitocentos e dezoito vezes inverso quinze igual a
22122003_12_05: setecentos e dezasseis mil quinhentos e treze virgula novecentos e dezasseis vezes quatro elevado a dez igual a
22122003_12_06: raiz indice sessenta a de cento e onze mil novecentos e dezassete virgula oitocentos e dezasseis igual a
22122003_12_07: duzentos e cinquenta e seis mil setecentos e onze virgula oitocentos e treze mais raiz cubica novecentos igual a
22122003_12_08: oitocentos e treze mil quinhentos e dezassete virgula setecentos e dezoito a dividir por cinquenta e sete ao cubo igual a
22122003_12_09: quinze mil duzentos e dez virgula quinhentos e onze menos apaga ultima mais raiz quadrada de quatro igual a
22122003_12_10: novecentos e doze mil oitocentos e quinze virgula cento e dezoito vezes dois ao quadrado igual a
22122003_12_11: duzentos e dezassete mil quinhentos e onze a dividir por seis elevado a cinco igual a
22122003_12_12: oitocentos e dez mil quinhentos e dezassete virgula duzentos e dez vezes treze ao cubo apaga tudo
22122003_12_13: cento e treze mil setecentos e doze virgula zero zero nove vezes trinta milhoes apaga ultima oitenta milhoes igual a
22122003_12_14: quatrocentos e dezassete mil cento e quinze virgula novecentos e treze a dividir por sete ao cubo igual a
22122003_12_15: setecentos e dez mil trezentos e dezoito mais raiz quadrada de oitenta e seis igual a
22122003_12_16: novecentos e dezasseis mil duzentos e dezassete virgula trezentos e doze a dividir por inverso de dez igual a
22122003_12_17: seiscentos oitenta e oito mil novecentos e vinte e tres virgula oitocentos e vinte sete menos um elevado zero igual a
22122003_12_18: seiscentos e onze mil trezentos e trinta e quatro virgula dezoito mais quarenta e nove ao cubo igual a
22122003_12_19: raiz indice sessenta a cubo de cem mil trezentos e trinta e seis virgula novecentos e dezasseis igual a
22122003_12_20: raiz quadrada de oitocentos e vinte sete mil setecentos e trinta e oito virgula quatrocentos e oitenta e quatro igual a
22122003_12_21: quinhentos e treze mil duzentos e quarenta e nove vezes abrir parenteses inverso vinte menos inverso de trinta fechar parenteses igual a
22122003_12_22: cem mil quinhentos cinquenta e nove virgula setecentos e dezanove a dividir por raiz quadrada de tres milhoes igual a
22122003_12_23: setecentos e dezoito mil e cem virgula novecentos e dezassete mais catorze ao cubo igual a
22122003_12_24: quatrocentos e treze mil quinhentos e cinquenta e seis virgula cem menos inverso de dezasseis milhoes igual a
22122003_12_25: duzentos e catorze mil cento e setenta e sete virgula cem a dividir por raiz quadrada quinze apaga tudo
22122003_13_01: setecentos oitenta e nove mil oitocentos e sessenta e cinco virgula seiscentos e noventa vezes raiz indice seis cem de trezentos igual a
22122003_13_02: oitocentos e dezanove mil quinhentos e setenta e cinco virgula quinhentos e dezanove menos sessenta e oito apaga tudo
22122003_13_03: quinhentos e onze mil oitocentos e setenta e tres virgula duzentos e doze um vezes sete elevado nove igual a
22122003_13_04: novecentos e cinquenta mil duzentos e noventa e oito virgula trezentos e quarenta e um a dividir por seis milhoes igual a
22122003_13_05: novecentos e doze mil cento e quinze a dividir por abrir parenteses cinco elevado a sete fechar parenteses igual a
22122003_13_06: cento e dezanove mil trezentos e onze virgula seiscentos e dez mais setenta elevado a vinte e dois igual a
22122003_13_07: cem mil quinhentos e dezanove virgula cem a dividir por apaga tudo raiz cubica de setecentos milhoes igual a
22122003_13_08: trezentos e treze mil novecentos e catorze virgula quinhentos e sessenta e tres menos oito ao quadrado igual a
22122003_13_09: cem mil duzentos e dezasseis virgula oitocentos e doze vezes apaga ultima mais raiz indice cinco de noventa e oito igual a
22122003_13_10: quatrocentos e doze mil trezentos e dezasseis virgula seiscentos e quarenta e seis a dividir por doze ao cubo apaga tudo
22122003_13_11: cento e dezasseis mil seiscentos e quinze virgula cem vezes apaga tudo raiz quadrada de oito a dividir por raiz cubica de nove igual a
22122003_13_12: oitocentos e noventa e nove mil trezentos e dezassete virgula cento e dezoito a dividir por zero ao quadrado igual a
22122003_13_13: duzentos e dezoito mil oitocentos e dez virgula setecentos e onze mais inverso de quinze elevado a quarenta igual a
22122003_13_14: cem mil novecentos e dezanove virgula setecentos e catorze a dividir por dezassete milhoes apaga tudo
22122003_13_15: quatrocentos e dezasseis mil quinhentos e treze virgula duzentos e noventa e cinco menos dois ao quadrado igual a
22122003_13_16: trezentos e dezassete mil quatrocentos e dezoito virgula cem vezes inverso de oitenta ao cubo igual a
22122003_13_17: cem mil seiscentos e treze virgula setecentos e vinte e nove a dividir por cinquenta e quatro elevado a zero igual a
22122003_13_18: quinhentos e quinze mil novecentos e dezoito virgula cento e doze vezes inverso de treze milhoes apaga ultima inverso de catorze milhoes igual a
22122003_13_19: quatrocentos e dezassete mil trezentos e quinze vezes abrir parenteses nove ao quadrado menos raiz indice seis de cem milhoes fechar parenteses igual a
22122003_13_20: setecentos e dez mil seiscentos e dezassete virgula novecentos e onze a dividir por sete ao cubo igual a
22122003_13_21: trezentos e dezoito mil duzentos e dez virgula zero doze mais sete ao quadrado igual a
22122003_13_22: cento e dezasseis mil oitocentos e dezanove a dividir por abrir parenteses sete ao cubo mais oito ao quadrado fechar parenteses igual a
22122003_13_23: seiscentos e dez mil setecentos e quinze virgula oitocentos e dezasseis menos catorze elevado a dez igual a
22122003_13_24: cem mil cento e dezanove virgula duzentos e noventa e sete vezes dezasseis elevado a nove apaga tudo
22122003_13_25: novecentos e onze mil quinhentos e dezoito virgula oitocentos e dezassete a dividir por raiz cubica de um milhao igual a
22122003_14_01: cem mil seiscentos e dezasseis virgula setecentos e treze vezes raiz indice quatro de oitocentos e noventa igual a
22122003_14_02: duzentos e quinze mil quatrocentos e onze virgula novecentos e doze vezes inverso cem igual a
22122003_14_03: quatrocentos e catorze mil trezentos e dezoito virgula cento e dezasseis a dividir por dois ao quadrado apaga ultima ao cubo igual a
22122003_14_04: quatrocentos e onze mil seiscentos e quinze virgula duzentos e dezoito mais treze milhoes elevado a dezanove igual a
22122003_14_05: novecentos e doze mil quatrocentos e dez virgula setecentos e dezanove a dividir por raiz cubica de duzentos igual a
22122003_14_06: cento e dezasseis mil novecentos e onze virgula seiscentos e doze menos apaga ultima mais inverso cinco milhoes igual a
22122003_14_07: duzentos e dezanove mil oitocentos e dez virgula trezentos e catorze vezes dois ao cubo igual a
22122003_14_08: seiscentos e dezoito mil quatrocentos e onze virgula quinhentos e dezoito a dividir por cinco elevado oito igual a
22122003_14_09: cento e treze mil quatrocentos e dezanove virgula novecentos e dez vezes quinze milhoes ao quadrado apaga tudo
22122003_14_10: quinhentos e vinte e sete mil seiscentos e treze virgula cem vezes inverso de oitenta e nove igual a
22122003_14_11: oitocentos e dezoito mil duzentos e quinze virgula cento e dezasseis a dividir por sete elevado a onze igual a
22122003_14_12: cento e dez mil quatrocentos e dezassete virgula seiscentos e onze mais catorze elevado a dezoito igual a
22122003_14_13: setecentos e onze mil cento e doze virgula duzentos e dezanove a dividir por raiz quadrada de setenta milhoes igual a
22122003_14_14: cento e dez mil e cem virgula oitocentos e doze menos abrir parenteses cinco ao cubo vezes tres ao quadrado fechar parenteses igual a
22122003_14_15: trezentos e dezanove mil quatrocentos e dezoito virgula setecentos e treze vezes inverso de cem igual a
22122003_14_16: duzentos e doze mil quinhentos e dezanove virgula seiscentos e onze a dividir por quinhentos milhoes apaga tudo
22122003_14_17: trezentos e doze mil oitocentos e dezoito virgula oitocentos e catorze vezes raiz indice treze de trezentos e sessenta e nove igual a

22122003_14_18: quatrocentos e treze mil cento e tres virgula trezentos e quinze vezes raiz cubica de dezassete igual a
22122003_14_19: quatrocentos e onze mil novecentos e dez virgula duzentos e dezanove a dividir por quinhentos e vinte sete apaga ultima cinquenta e seis igual a
22122003_14_20: seiscentos e doze mil setecentos e quinze virgula cento e dezoito mais nove elevado a zero igual a
22122003_14_21: novecentos e dez mil quatrocentos e dezanove virgula duzentos e doze a dividir por inverso de cento e setenta igual a
22122003_14_22: oitocentos e dezassete mil quatrocentos e dezoito virgula quatrocentos e dez menos inverso setenta igual a
22122003_14_23: setecentos e onze mil novecentos e dezasseis virgula oitocentos e dezassete vezes dezasseis elevado a nove igual a
22122003_14_24: quatrocentos e dezanove mil e cem virgula duzentos oitenta e sete a dividir por trezentos milhoes igual a
22122003_14_25: trezentos e dezanove mil quatrocentos e doze virgula cento e onze vezes raiz indice oitenta de quatrocentos e trinta e um igual a
22122003_15_01: quatrocentos e dezanove mil novecentos e doze virgula oitocentos e quinze vezes inverso dois igual a
22122003_15_02: setecentos e treze mil seiscentos e dezassete virgula zero dezanove a dividir por tres ao cubo igual a
22122003_15_03: cento e dezassete mil quatrocentos e quinze virgula seiscentos e dezoito mais quinze milhoes ao quadrado igual a
22122003_15_04: quinhentos e onze mil oitocentos e doze virgula quinhentos e dezasseis a dividir por raiz cubica de trinta e dois igual a
22122003_15_05: novecentos e doze mil e cem virgula setecentos e noventa e dois menos raiz quadrada de dezasseis igual a
22122003_15_06: cento e dezoito mil quatrocentos e cinquenta e cinco virgula cem vezes cem elevado a quatro igual a
22122003_15_07: duzentos e dezasseis mil trezentos e dezasseis virgula trezentos e onze a dividir por apaga tudo
22122003_15_08: oito milhoes menos dez mil igual a
22122003_15_09: cento e quinze mil quinhentos e dez virgula oitocentos e dezoito vezes raiz indice dezanove de seiscentos e quarenta e um apaga tudo
22122003_15_10: setecentos e doze mil quatrocentos e onze vezes abrir parenteses nove ao cubo menos inverso quinze fechar parenteses igual a
22122003_15_11: duzentos e dezasseis mil setecentos e dezassete virgula seiscentos e dezanove a dividir por quarenta e sete elevado oito igual a
22122003_15_12: seiscentos e dezassete mil cento e dez virgula cento e catorze mais raiz quadrada de catorze igual a
22122003_15_13: seiscentos e dezassete mil cento e dez virgula cento e catorze mais raiz quadrada de catorze igual a
22122003_15_14: quinhentos e dezanove mil duzentos e catorze virgula quinhentos e doze a dividir por seis ao cubo igual a
22122003_15_15: novecentos e dezoito mil oitocentos e onze virgula zero catorze menos treze ao quadrado igual a
22122003_15_16: novecentos e doze mil novecentos e quinze virgula cento e doze vezes raiz quadrada de trezentos e vinte igual a
22122003_15_17: trezentos e dez mil seiscentos e dezanove a dividir por dezassete elevado a seis igual a
22122003_15_18: quatrocentos e doze mil trezentos e dezasseis virgula duzentos e dezassete menos raiz quadrada de trinta e tres igual a
22122003_15_19: cento e dezasseis mil oitocentos e dezoito virgula seiscentos e quinze vezes cinco elevado a cinco apaga tudo
22122003_15_20: setecentos e onze mil cento e dezassete virgula duzentos e dezanove a dividir por inverso sete igual a
22122003_15_21: novecentos e setenta e dois mil duzentos e cinquenta e dois virgula cem mais cinco milhoes igual a
22122003_15_22: trezentos e dez mil duzentos e noventa e tres virgula cem a dividir por inverso de treze elevado a seis igual a
22122003_15_23: quinhentos e dezasseis mil duzentos e onze virgula setecentos e dez menos dezoito elevado nove igual a
22122003_15_24: cento e dezassete mil duzentos e dezassete virgula quinhentos e dezoito vezes raiz cubica de vinte e sete igual a
22122003_15_25: setecentos e doze mil cento e treze virgula trezentos e dezasseis a dividir por quinze elevado a nove igual a
22122003_16_01: quinhentos e catorze mil setecentos e dezasseis virgula novecentos e onze vezes dezasseis ao cubo apaga tudo
22122003_16_02: setecentos e dezoito mil oitocentos e dezassete virgula cento e onze mais inverso cem igual a
22122003_16_03: duzentos e dezasseis mil trezentos e treze virgula oitocentos e nove a dividir por sete ao cubo apaga ultima ao quadrado igual a
22122003_16_04: novecentos e onze mil cento e dezassete apaga ultima dezanove mais inverso treze igual a
22122003_16_05: quinhentos e dezassete mil cento e dezasseis virgula duzentos e doze a dividir por seis ao cubo igual a
22122003_16_06: quatrocentos e doze mil setecentos e dezoito virgula setecentos e catorze menos vinte ao quadrado apaga tudo
22122003_16_07: oitocentos e catorze mil duzentos e dezasseis vezes abrir parenteses duzentos e um menos quatrocentos e tres fechar parenteses igual a
22122003_16_08: duzentos e onze mil e cem virgula novecentos e quinze a dividir por tudo oito elevado a dezassete apaga tudo
22122003_16_09: trezentos e doze mil cento e quinze virgula oitocentos e dezoito vezes inverso quinze igual a
22122003_16_10: setecentos e dezasseis mil quinhentos e treze virgula novecentos e dezasseis vezes quatro elevado a dez igual a
22122003_16_11: raiz indice sessenta a de cento e onze mil novecentos e dezassete virgula oitocentos e dezasseis igual a
22122003_16_12: duzentos e cinquenta e seis mil setecentos e onze virgula oitocentos e treze mais raiz cubica de novecentos igual a
22122003_16_13: oitocentos e treze mil quinhentos e dezassete virgula setecentos e dezoito a dividir por cinquenta e sete ao cubo igual a
22122003_16_14: quinze mil duzentos e dez virgula quinhentos e onze menos apaga ultima mais raiz quadrada de quatro igual a
22122003_16_15: novecentos e doze mil oitocentos e quinze virgula cento e dezoito vezes dois ao quadrado igual a
22122003_16_16: duzentos e dezassete mil quinhentos e onze a dividir por seis elevado a cinco igual a
22122003_16_17: oitocentos e dez mil quinhentos e dezassete virgula duzentos e dez vezes treze ao cubo apaga tudo
22122003_16_18: cento e treze mil setecentos e doze virgula zero zero nove vezes trinta milhoes apaga ultima oitocentos milhoes igual a
22122003_16_19: quatrocentos e dezassete mil cento e quinze virgula novecentos e treze a dividir por sete ao cubo igual a
22122003_16_20: setecentos e dez mil trezentos e dezoito mais raiz quadrada de oitenta e seis igual a
22122003_16_21: novecentos e dezasseis mil duzentos e dezassete virgula trezentos e doze a dividir por inverso de dez igual a
22122003_16_22: seiscentos e oitenta e oito mil novecentos e vinte e tres virgula oitocentos e vinte sete menos um elevado a zero igual a
22122003_16_23: seiscentos e onze mil trezentos e trinta e quatro virgula dezoito mais quarenta e nove ao cubo igual a
22122003_16_24: raiz indice sessenta de cem mil trezentos e trinta e seis virgula novecentos e dezasseis igual a
22122003_16_25: raiz quadrada de oitocentos e vinte e sete mil setecentos e trinta e oito virgula quatrocentos e oitenta e quatro igual a
22122003_17_01: quinhentos e treze mil duzentos e quarenta e nove vezes abrir parenteses inverso vinte menos inverso trinta fechar parenteses igual a
22122003_17_02: cem mil quinhentos e cinquenta e nove virgula setecentos e dezanove a dividir por raiz quadrada de tres milhoes igual a
22122003_17_03: setecentos e dezoito mil e cem virgula novecentos e dezassete mais catorze ao cubo igual a
22122003_17_04: quatrocentos e treze mil quinhentos e cinquenta e seis virgula cem menos inverso de dezasseis milhoes igual a
22122003_17_05: duzentos e catorze mil cento e sessenta e sete virgula cem a dividir por raiz quadrada de quinze apaga tudo
22122003_17_06: setecentos e oitenta e nove mil oitocentos e sessenta e cinco virgula seiscentos e noventa vezes raiz indice seis de trezentos igual a
22122003_17_07: oitocentos e dezanove mil quinhentos e setenta e cinco virgula quinhentos e dezanove menos sessenta e oito apaga tudo
22122003_17_08: quinhentos e onze mil oitocentos e sessenta e tres virgula duzentos e doze vezes sete elevado a nove igual a
22122003_17_09: novecentos e cinquenta mil duzentos e noventa e oito virgula trezentos e quarenta e um a dividir por seis milhoes igual a
23122003_17_10: zero virgula seiscentos e catorze ao quadrado a dividir por um milhao oitocentos e sessenta e nove apaga tudo
23122003_17_11: zero virgula trezentos e sessenta e nove menos inverso cinco virgula catorze igual a
23122003_17_12: abrir parenteses um virgula oitocentos e catorze menos trinta e seis fechar parenteses ao cubo igual a
23122003_17_13: raiz indice dez de quarenta e cinco mil cento e catorze virgula cem vezes oitenta e oito ao cubo igual a
23122003_17_14: raiz quadrada de abrir parenteses dezassete mil duzentos e setenta e cinco virgula cem menos dois ao cubo fechar parenteses apaga tudo
23122003_17_15: raiz indice cinco de cento e dez mil oitocentos e dezoito virgula quatrocentos e cinquenta e oito apaga ultima
23122003_17_16: abrir parenteses zero virgula seiscentos e dezassete mais zero virgula quarenta e seis fechar parenteses a dividir por tres ao cubo igual a
23122003_17_17: inverso de zero virgula quatrocentos e treze vezes apaga ultima raiz cubica de trinta ao quadrado igual a
23122003_17_18: zero virgula setecentos e catorze apaga tudo
23122003_17_19: cinco ao cubo menos nove ao quadrado igual a
23122003_17_20: raiz indice noventa de abrir parenteses um virgula zero nove ao cubo mais inverso cinquenta e sete apaga tudo
23122003_17_21: zero virgula cento e treze menos dois virgula trinta e cinco menos zero ao quadrado apaga tudo
23122003_17_22: abrir parenteses um virgula oitocentos e catorze vezes tres ao quadrado fechar parenteses ao cubo igual a
23122003_17_23: raiz cubica de um virgula trezentos e doze elevado a menos quatro apaga tudo
23122003_17_24: raiz quadrada de inverso setenta e oito igual a

23122003_17_25: raiz indice dez de quarenta e seis mil cento e doze virgula trezentos e sessenta e dois elevado a seis apaga ultima igual a
23122003_18_01: abrir parenteses zero virgula setecentos e oito mais sete virgula vinte e tres fechar parenteses a dividir por inverso de um milhao igual a
23122003_18_02: raiz cubica de zero virgula novecentos e dezasseis mais raiz indice oito de quarenta e um apaga tudo
23122003_18_03: logaritmo base dois de mil seiscentos e onze virgula quatrocentos e oitenta e seis a dividir por raiz quadrada de sete milhoes igual a
23122003_18_04: zero virgula duzentos e noventa e seis apaga tudo
23122003_18_05: zero virgula duzentos e noventa e seis apaga ultima raiz cubica de sessenta e oito ao quadrado igual a
23122003_18_06: abrir parenteses um virgula oitocentos e dez menos raiz cubica vinte e sete fechar parenteses vezes logaritmo base dez de cem milhoes igual a
23122003_18_07: raiz indice nove de zero virgula novecentos e treze mais raiz quadrada de dezasseis igual a
23122003_18_08: zero virgula quatrocentos e dezasseis apaga tudo um ao cubo menos inverso tres igual a
23122003_18_09: zero virgula oitocentos e catorze elevado a setenta e oito igual a
23122003_18_10: um virgula duzentos e quinze menos abrir parenteses raiz cubica de sessenta e seis vezes apaga ultima mais quatro ao cubo fechar ao parenteses igual a
23122003_18_11: logaritmo base nove de oitocentos e noventa e cinco mil e quinze virgula oitocentos e catorze apaga tudo
23122003_18_12: zero virgula trezentos e treze apaga ultima catorze a dividir por um milhao elevado a menos dois igual a
23122003_18_13: logaritmo base cinco de mil duzentos e dez menos raiz cubica de oitenta e tres ao quadrado igual a
23122003_18_14: um apaga ultima um milhao quatrocentos e onze vezes raiz quadrada de dezasseis igual a
23122003_18_15: abrir parenteses mil quatrocentos e dezoito menos um milhao e seiscentos e trinta mil fechar parenteses vezes raiz indice sete de cinquenta e oito igual a
23122003_18_16: mil novecentos e dezasseis apaga ultima logaritmo base dez de cento e catorze mil ao cubo apaga tudo
23122003_18_17: raiz cubica de dez milhoes trezentos e setenta e sete virgula quatrocentos e dezasseis igual a
23122003_18_18: raiz quadrada de treze mil oitocentos e noventa e quatro virgula cento e dezanove apaga tudo
23122003_18_19: mil setecentos e dezassete mais abrir parenteses tres ao quadrado menos logaritmo de trinta e dois fechar parenteses igual a
23122003_18_20: um milhao seiscentos e dezanove mil e quarenta e cinco mais sessenta e um ao cubo igual a
23122003_18_21: mil quinhentos e catorze a dividir por abrir parenteses raiz quadrada de setenta menos raiz cubica de cinquenta fechar parenteses igual a
23122003_18_22: logaritmo base dois de um milhao oitocentos e doze apaga ultima setecentos e sete mil igual a
23122003_18_23: abrir parenteses a um novecentos e dezoito mais cinco ao quadrado fechar parenteses apaga tudo
23122003_18_24: mil trezentos e quinze mais abrir parenteses dois ao cubo menos tres ao quadrado fechar parenteses igual a
23122003_18_25: mil seiscentos e doze apaga ultima raiz cubica de inverso quarenta e um apaga tudo
23122003_19_01: logaritmo base quatro de mil trezentos e dezasseis virgula seiscentos e catorze igual a
23122003_19_02: abrir parenteses raiz cubica de mil duzentos e dezassete fechar parenteses ao quadrado igual a
23122003_19_03: inverso de mil quinhentos e onze virgula quinhentos e tres apaga ultima dois igual a
23122003_19_04: logaritmo base dez de mil setecentos e catorze vezes raiz cubica de sessenta e dois igual a
23122003_19_05: um milhao quatrocentos e quinze virgula zero catorze apaga tudo
23122003_19_06: nove ao quadrado mais oito ao quadrado igual a
23122003_19_07: raiz indice oitenta de trezentos e quarenta e nove milhoes quatrocentos e dezasseis virgula cem apaga tudo
23122003_19_08: logaritmo base nove de oitenta e tres mil setecentos e trinta e nove virgula cento e quinze igual a
23122003_19_09: raiz cubica de um milhao cento e treze virgula oitocentos e treze menos dois elevado a treze igual a
23122003_19_10: abrir parenteses mil seiscentos e catorze ao quadrado vezes sessenta e treze ao cubo fechar parenteses igual a
23122003_19_11: logaritmo base cinco de um milhao duzentos e dezanove a dividir por apaga ultima vezes noventa ao cubo igual a
23122003_19_12: abrir parenteses mil trezentos e dezassete ao cubo a dividir por sete milhoes ao quadrado apaga ultima elevado a quatro fechar parenteses igual a
23122003_19_13: mil novecentos e dezasseis virgula novecentos e catorze apaga ultima
23122003_19_14: mil novecentos e dezasseis virgula novecentos e catorze apaga tudo um milhao elevado a sete igual a
23122003_19_15: abrir parenteses inverso de mil oitocentos e dezassete menos inverso tres milhoes fechar parenteses vezes sessenta e nove ao quadrado igual a
23122003_19_16: raiz quadrada de mil cento e dezoito apaga ultima apaga tudo raiz cubica de cinquenta e nove ao quadrado igual a
23122003_19_17: raiz indice cinco de logaritmo base tres de mil setecentos e quinze apaga ultima trinta e oito apaga tudo
23122003_19_18: raiz cubica de mil quinhentos e onze virgula zero zero sete elevado a menos cinco apaga ultima quatro igual a
23122003_19_19: abrir parenteses raiz quadrada oito a dividir por raiz cubica de setenta e dois fechar parenteses igual a
23122003_19_20: abrir parenteses mil cento e dez mais noventa e quatro ao cubo fechar parenteses vezes inverso vinte apaga tudo
23122003_19_21: setecentos milhoes trinta e sete mil quatrocentos e dezanove virgula quatrocentos e dez apaga tudo
23122003_19_22: raiz cubica de noventa e sete igual a
23122003_19_23: mil setecentos e onze virgula novecentos e trinta e seis ao quadrado vezes logaritmo de um milhao igual a
23122003_19_24: logaritmo de abrir parenteses mil duzentos e dezassete ao cubo mais oito virgula vinte e tres fechar parenteses igual a
23122003_19_25: raiz cubica de um milhao quatrocentos e dez virgula quinhentos e trinta e quatro apaga tudo
23122003_20_01: raiz indice dois de trinta e quatro mais raiz indice seis de sessenta e um apaga tudo
23122003_20_02: abrir parenteses logaritmo base dez de apaga ultima raiz quadrada seis ao quadrado igual a
23122003_20_03: raiz cubica de apaga ultima abrir parenteses raiz indice trinta de zero virgula dois fechar parenteses ao quadrado igual a
23122003_20_04: logaritmo base cinco de abrir parenteses raiz quadrada de dois mais raiz indice quatro de oito fechar parenteses igual a
23122003_20_05: raiz cubica de setenta ao cubo a dividir por inverso de apaga ultima logaritmo de um igual a
23122003_20_06: raiz indice oito de raiz quadrada cinco ao quadrado vezes logaritmo base um de oitenta apaga tudo
23122003_20_07: logaritmo base trinta abrir parenteses tres ao quadrado menos quarenta ao quadrado fechar parenteses igual a
23122003_20_08: raiz quadrada de apaga ultima raiz indice doze de zero virgula vinte e nove igual a
23122003_20_09: raiz cubica de quatro virgula sessenta e seis igual a
23122003_20_10: raiz cubica apaga ultima logaritmo base seis de raiz quadrada de catorze igual a
23122003_20_11: abrir parenteses logaritmo base dois quatro a dividir por raiz indice oito de dezasseis fechar parenteses igual a
23122003_20_12: raiz quadrada cem menos raiz cubica de um milhao apaga ultima nove milhoes igual a
23122003_20_13: logaritmo base dez de um virgula sete a dividir por raiz quadrada de tres virgula noventa e quatro apaga tudo
23122003_20_14: raiz indice dois de apaga ultima raiz quadrada de sessenta e um mais raiz cubica de setenta e quatro igual a
23122003_20_15: abrir parenteses raiz quadrada de trinta e dois mais raiz indice seis de quarenta e sete fechar parenteses igual a
23122003_20_16: logaritmo base dois de um milhao e cem mil mais raiz cubica de oitenta milhoes apaga ultima noventa milhoes igual a
23122003_20_17: abrir parenteses raiz quadrada de setenta e dois menos raiz cubica de vinte e nove fechar parenteses igual a
23122003_20_18: um milhao apaga ultima dois milhoes vezes raiz indice nove de logaritmo base dez de cinquenta e seis igual a
23122003_20_19: raiz indice cinco de doze virgula vinte e sete ao quadrado apaga ultima igual a
23122003_20_20: logaritmo base dez de raiz quadrada vinte menos raiz cubica de oitenta e oito apaga ultima igual a
23122003_20_21: raiz quadrada de quarenta e dois a dividir por raiz cubica de vinte e tres igual a
23122003_20_22: raiz indice sete de logaritmo base dois de setenta e nove igual a
23122003_20_23: logaritmo base vinte de raiz quadrada quatro mais raiz cubica nove igual a
23122003_20_24: raiz indice cinco de logaritmo base tres de quarenta e oito vezes raiz quadrada de zero virgula um igual a
23122003_20_25: raiz cubica de zero virgula vinte e sete mais logaritmo base dois de um virgula quatro igual a
23122003_21_01: logaritmo base dois de abrir parenteses um ao quadrado mais raiz quadrada de noventa e tres fechar parenteses igual a
23122003_21_02: raiz cubica de um milhao e oitenta mil vezes raiz quadrada de seis milhoes e trinta e nove mil apaga tudo
23122003_21_03: raiz indice quatro de zero virgula sessenta e oito a dividir por raiz indice seis de cinco virgula sessenta e tres igual a
23122003_21_04: logaritmo base dois de raiz indice tres de vinte e um milhoes igual a
23122003_21_05: raiz cubica nove menos raiz quadrada de quatro mais logaritmo base dois de oito igual a
23122003_21_06: logaritmo base dez de cinquenta e nove a dividir por raiz indice oito de sessenta e quatro igual a

23122003_21_07: raiz quadrada de sessenta e cinco um menos raiz cubica de oitenta e nove igual a
23122003_21_08: logaritmo base sete de cinquenta e tres mais logaritmo base nove de trinta e seis apaga tudo
23122003_21_09: raiz indice menos dois de um milhao e quarenta mil a dividir por logaritmo base cinco de vinte milhoes igual a
23122003_21_10: raiz cubica de um virgula dois vezes raiz indice cinco de dez milhoes e setenta mil igual a
23122003_21_11: raiz quadrada de cinquenta e nove vezes raiz cubica de sessenta e oito igual a
23122003_21_12: logaritmo base dois de quarenta milhoes ao cubo mais raiz indice sete de um milhao e oitenta e mil apaga tudo
23122003_21_13: raiz quadrada de um milhao ao quadrado mais inverso de apaga ultima logaritmo de tres milhoes igual a
23122003_21_14: raiz indice seis de setenta e nove milhoes igual a
23122003_21_15: logaritmo base quatro de cinquenta e um milhoes mais um milhao e oitenta igual a
23122003_21_16: zero virgula seiscentos e catorze ao quadrado a dividir por um milhao oitocentos e sessenta e nove apaga tudo
23122003_21_17: zero virgula trezentos e sessenta e nove menos inverso cinco virgula catorze igual a
23122003_21_18: abrir parenteses um virgula oitocentos e catorze menos trinta e seis fechar parenteses ao cubo igual a
23122003_21_19: raiz indice dez de quarenta e cinco mil cento e catorze virgula cem vezes oitenta e oito ao cubo igual a
23122003_21_20: raiz quadrada de abrir parenteses dezassete mil duzentos e setenta e cinco virgula cem menos dois ao cubo fechar parenteses apaga tudo
23122003_21_21: raiz indice cinco de cento e dez mil oitocentos e dezoito virgula quatrocentos e cinquenta e oito apaga ultima
23122003_21_22: abrir parenteses zero virgula seiscentos e dezassete mais zero virgula quarenta e seis fechar parenteses a dividir por tres ao cubo igual a
23122003_21_23: inverso de zero virgula quatrocentos e treze vezes apaga ultima raiz quadrada de trinta ao quadrado igual a
23122003_21_24: zero virgula setecentos e catorze apaga tudo
23122003_21_25: cinco ao cubo menos nove ao quadrado igual a
23122003_22_01: raiz indice noventa abrir parenteses um virgula zero nove ao cubo mais inverso cinquenta e sete apaga tudo
23122003_22_02: zero virgula cento e treze menos dois virgula cem trinta e cinco menos zero ao quadrado apaga tudo
23122003_22_03: abrir parenteses um virgula oitocentos e catorze vezes tres ao quadrado fechar parenteses ao cubo igual a
23122003_22_04: raiz cubica de um virgula trezentos e doze elevado a menos quatro apaga tudo
23122003_22_05: raiz quadrada de inverso de setenta e oito igual a
23122003_22_06: raiz indice dez de quarenta e seis mil cento e doze virgula trezentos e sessenta e dois elevado a seis apaga ultima igual a
23122003_22_07: abrir parenteses zero virgula setecentos e oito mais sete virgula vinte e tres fechar parenteses a dividir por inverso de um milhao igual a
23122003_22_08: raiz cubica de zero virgula novecentos e dezasseis mais raiz indice oito de quarenta e um apaga tudo
23122003_22_09: logaritmo base dois de mil seiscentos e onze virgula quatrocentos e oitenta e seis a dividir por raiz quadrada sete milhoes igual a
23122003_22_10: zero virgula duzentos e noventa e seis apaga ultima raiz cubica de sessenta e oito ao quadrado igual a
23122003_22_11: abrir parenteses um virgula oitocentos e dez menos raiz cubica de vinte e sete fechar parenteses vezes logaritmo base dez cem milhoes igual a
23122003_22_12: raiz indice nove de zero virgula novecentos e treze mais raiz quadrada de dezasseis igual a
23122003_22_13: zero virgula quatrocentos e dezasseis apaga tudo
23122003_22_14: um ao cubo menos inverso tres igual a
23122003_22_15: zero virgula oitocentos e catorze elevado a setenta e oito igual a
23122003_22_16: um virgula duzentos e quinze menos abrir parenteses raiz cubica de sessenta e seis vezes apaga ultima mais quatro ao um a ao cubo
23122003_22_17: um virgula duzentos e quinze menos abrir parenteses raiz cubica de sessenta e seis vezes apaga ultima mais quatro ao cubo fechar parenteses igual a
23122003_22_18: logaritmo base nove de oitocentos e noventa e cinco mil e quinze virgula oitocentos e catorze apaga tudo
23122003_22_19: zero virgula trezentos e treze apaga ultima catorze a dividir por um milhao elevado a menos dois igual a
23122003_22_20: logaritmo base cinco de mil duzentos e dez menos raiz cubica de oitenta e tres ao quadrado igual a
23122003_22_21: um apaga ultima um milhao quatrocentos e onze vezes raiz quadrada de dezasseis igual a
23122003_22_22: abrir parenteses mil quatrocentos e dezoito menos um milhao e seiscentos e trinta mil fechar parenteses vezes raiz indice sete de cinquenta e oito igual a
23122003_22_23: mil novecentos e dezasseis apaga ultima logaritmo base dez de cento e catorze mil ao cubo apaga tudo
23122003_22_24: raiz cubica de dez milhoes trezentos e setenta e sete virgula quatrocentos e dezasseis igual a
23122003_22_25: raiz quadrada de treze mil oitocentos e noventa e quatro virgula cento e dezanove apaga tudo
23122003_23_01: mil setecentos e dezassete mais abrir parenteses tres ao quadrado menos logaritmo de trinta e dois fechar parenteses igual a
23122003_23_02: um milhao seiscentos e dezanove mil e quarenta e cinco mais sessenta e um ao cubo igual a
23122003_23_03: mil quinhentos e catorze a dividir por abrir parenteses raiz quadrada de setenta menos raiz cubica de cinquenta fechar parenteses igual a
23122003_23_04: logaritmo base dois de um milhao oitocentos e doze apaga ultima setecentos e sete mil igual a
23122003_23_05: abrir parenteses mil novecentos e dezoito mais cinco ao quadrado fechar parenteses apaga tudo
23122003_23_06: mil trezentos e quinze mais abrir parenteses dois ao cubo menos tres ao quadrado fechar parenteses igual a
23122003_23_07: mil seiscentos e doze apaga ultima raiz cubica de inverso de quarenta e um apaga tudo
23122003_23_08: logaritmo base quatro de mil trezentos e dezasseis virgula seiscentos e catorze igual a
23122003_23_09: abrir parenteses raiz cubica de mil duzentos e dezassete fechar parenteses ao quadrado igual a
23122003_23_10: inverso de mil quinhentos e onze virgula quinhentos e tres apaga ultima dois igual a
23122003_23_11: logaritmo base dez de mil setecentos e catorze vezes raiz cubica de sessenta e dois igual a
23122003_23_12: um milhao quatrocentos e quinze virgula zero catorze apaga tudo
23122003_23_13: um nove ao cubo mais oito ao quadrado igual a
23122003_23_14: raiz indice oitenta de trezentos e quarenta e nove milhoes quatrocentos e dezasseis virgula cem apaga tudo
23122003_23_15: logaritmo base nove de oitenta e tres mil setecentos e trinta e nove virgula cento e quinze igual a
23122003_23_16: raiz cubica de um milhao cento e treze virgula oitocentos e treze menos dois elevado a treze igual a
23122003_23_17: abrir parenteses mil seiscentos e catorze ao quadrado vezes sessenta e tres ao cubo fechar parenteses igual a
23122003_23_18: logaritmo base cinco de um milhao duzentos e dezanove a dividir por apaga ultima vezes noventa ao cubo igual a
23122003_23_19: abrir parenteses mil trezentos e dezassete ao cubo a dividir por sete milhoes ao quadrado apaga ultima elevado a quatro fechar parenteses igual a
23122003_23_20: mil novecentos e dezasseis virgula novecentos e catorze apaga tudo
23122003_23_21: um milhao elevado a sete igual a
23122003_23_22: abrir parenteses de inverso de mil oitocentos e dezassete menos inverso de tres milhoes fechar parenteses vezes setenta e nove ao quadrado igual a
23122003_23_23: raiz quadrada de mil cento e dezoito apaga ultima apaga tudo
23122003_23_24: raiz cubica de cinquenta e nove ao quadrado igual a
23122003_23_25: raiz indice cinco de logaritmo base tres de mil setecentos e quinze apaga ultima trinta e oito apaga tudo
23122003_24_01: raiz cubica de mil quinhentos e onze virgula zero zero sete elevado a menos cinco apaga ultima quatro igual a
23122003_24_02: abrir parenteses raiz quadrada oito a dividir por raiz cubica de setenta e dois fechar parenteses igual a
23122003_24_03: abrir parenteses mil cento e dez mais noventa e quatro ao cubo fechar parenteses vezes inverso vinte apaga tudo
23122003_24_04: setecentos milhoes trinta e sete mil quatrocentos e dezanove virgula quatrocentos e dez apaga tudo
23122003_24_05: raiz cubica de noventa e sete igual a
23122003_24_06: mil setecentos e onze virgula novecentos e trinta e seis ao quadrado vezes logaritmo de um milhao igual a
23122003_24_07: logaritmo de abrir parenteses mil duzentos e dezassete ao cubo mais oito virgula vinte e tres fechar parenteses igual a
23122003_24_08: raiz cubica de um milhao quatrocentos e dez virgula quinhentos e trinta e quatro apaga tudo
23122003_24_10: abrir parenteses logaritmo base dez de apaga ultima raiz quadrada de seis ao quadrado igual a
23122003_24_11: raiz cubica de apaga ultima abrir parenteses raiz indice trinta de zero virgula dois fechar parenteses ao quadrado igual a
23122003_24_12: logaritmo base cinco de abrir parenteses raiz quadrada dois mais raiz indice quatro de oito fechar parenteses igual a
23122003_24_13: raiz cubica de setenta ao cubo a dividir por inverso de apaga ultima logaritmo e um igual a
23122003_24_14: raiz indice oito de raiz quadrada cinco ao quadrado vezes logaritmo base de um de oitenta apaga tudo

23122003_24_15: logaritmo base trinta de abrir parenteses tres ao quadrado menos quarenta ao quadrado fechar parenteses igual a
23122003_24_16: raiz quadrada de apaga ultima raiz indice doze de zero virgula vinte e nove igual a
23122003_24_17: raiz cubica e quatro virgula sessenta e seis igual a
23122003_24_18: raiz cubica apaga ultima logaritmo base seis de raiz quadrada de catorze igual a
23122003_24_19: abrir parenteses logaritmo base dois de quatro a dividir por raiz indice oito de dezasseis fechar parenteses igual a
23122003_24_20: raiz quadrada cem menos raiz cubica de um milhao apaga ultima nove milhoes igual a
23122003_24_21: logaritmo base dez de um virgula sete a dividir por raiz quadrada de tres virgula noventa e quatro apaga tudo
23122003_24_22: raiz indice dois de apaga ultima raiz quadrada de sessenta e um mais raiz cubica de setenta e quatro igual a
23122003_24_23: abrir parenteses raiz quadrada de trinta e dois mais raiz indice seis de quarenta e sete fechar parenteses igual a
23122003_24_24: logaritmo base dois de um milhao e cem mil mais raiz cubica de oitenta milhoes apaga ultima noventa milhoes igual a
23122003_24_25: abrir parenteses raiz quadrada de setenta e dois menos raiz cubica de vinte e nove fechar parenteses igual a
23122003_25_01: um milhao apaga ultima dois milhoes vezes raiz indice nove de logaritmo base dez de cinquenta e seis igual a
23122003_25_02: raiz indice cinco de doze virgula vinte e sete ao quadrado apaga ultima igual a
23122003_25_03: logaritmo base dez de raiz quadrada vinte menos raiz cubica de oitenta e oito apaga ultima igual a
23122003_25_04: raiz quadrada de quarenta e dois a dividir por raiz cubica de vinte e tres igual a
23122003_25_05: raiz indice sete de logaritmo base dois de setenta e nove igual a
23122003_25_06: logaritmo base vinte de raiz quadrada de quatro mais raiz cubica nove igual a
23122003_25_07: raiz indice cinco de logaritmo base tres de quarenta e oito vezes raiz quadrada de zero virgula um igual a
23122003_25_08: raiz cubica de zero virgula vinte e sete mais logaritmo base dois de um virgula quatro igual a
23122003_25_09: logaritmo base dois de abrir parenteses um ao quadrado mais raiz quadrada de noventa e tres fechar parenteses igual a
23122003_25_10: raiz cubica de um milhao e oitenta e mil vezes raiz quadrada de seis milhoes e trinta e nove mil apaga tudo
23122003_25_11: raiz indice quatro de zero virgula sessenta e oito a dividir por raiz indice seis de cinco virgula sessenta e tres igual a
23122003_25_12: logaritmo base dois de raiz indice tres de vinte e um milhoes igual a
23122003_25_13: raiz cúbica de nove menos raiz quadrada de quatro mais logaritmo base dois oito igual a
23122003_25_14: logaritmo base dez de cinquenta e nove a dividir por raiz indice oito de sessenta e quatro igual a
23122003_25_15: raiz quadrada de sessenta e cinco menos raiz cubica de oitenta e nove igual a
23122003_25_16: logaritmo base sete de cinquenta e tres mais logaritmo base nove de trinta e seis apaga tudo
23122003_25_17: raiz indice menos dois de um milhao e quarenta mil a dividir por logaritmo base cinco de vinte milhoes igual a
23122003_25_18: raiz cubica de um virgula dois vezes raiz indice cinco de dez milhoes e setenta mil igual a
23122003_25_19: raiz quadrada de cinquenta e nove vezes raiz cubica de sessenta e oito igual a
23122003_25_20: logaritmo base dois de quarenta milhoes ao cubo mais raiz indice sete de um milhao e oitenta mil apaga tudo
23122003_25_21: raiz quadrada de um milhao ao quadrado mais inverso de apaga ultima logaritmo de tres milhoes igual a
23122003_25_22: raiz indice seis de sessenta e nove milhoes igual a
23122003_25_23: logaritmo base quatro de cinquenta e um mais um milhao e oitenta igual a
23122003_25_24: logaritmo base dois de mil seiscentos e onze virgula quatrocentos e oitenta e seis a dividir por raiz quadrada de
23122003_25_25: raiz indice oito de raiz quadrada de cinco ao quadrado vezes logaritmo de oitenta igual a
24112003_26_01: duzentos e doze mil seiscentos e dezasseis virgula oitocentos e doze mais abrir parenteses raiz indice cem de setecentos
24112003_26_02: duzentos e doze mil seiscentos e dezasseis virgula oitocentos e doze mais abrir parenteses raiz indice vinte e sete um
24112003_26_03: quinhentos e dez mil
24112003_26_04: novecentos onze mil seiscentos e dezoito virgula novecentos e dezanove
24112003_26_05: trezentos e dezanove mil setecentos e dezoito virgula novecentos e dezassete mais raiz indice sete de cem igual a
24112003_26_06: dezanove mil quatrocentos e cem
24112003_26_07: mil quatrocentos e dez elevado a cinquenta e seis mais logaritmo base dez de inverso sete apaga ultima logaritmo de dezassete milhoes igual a
24112003_26_08: um milhao oitocentos e treze mil seiscentos e quinze menos raiz indice cinquenta de setecentos e vinte mil apaga tudo
24112003_26_09: cinco milhoes setecentos e dezasseis mil oitocentos e dezoito virgula novecentos e dezanove menos apaga ultima a dividir por raiz quadrada de trinta e sete elevado oito igual a
22122003_26_10: raiz quadrada de sessenta e dois mil quinhentos e sessenta a
22122003_26_11: novecentos e dez mil quatrocentos e dezanove virgula duzentos e doze a dividir por inverso de setenta
22122003_26_12: seiscentos e oitenta e oito mil novecentos e vinte mais tres virgula oitocentos e vinte sete
23122003_26_13: um virgula duzentos e quinze menos abrir parenteses raiz cubica de sessenta e seis vezes apaga ultima mais quatro ao cubo

ANEXO B

Estruturação das frases da LPFAV2 em conjunto de treino e teste

Foi realizada a distribuição das frases pelos grupos de modo a tentar garantir um número mínimo de amostras de cada palavra por grupo.

Conjunto de frases para semear os modelos:

24112003_01_01 24112003_01_02 24112003_01_03 24112003_01_04 24112003_01_05 24112003_01_06 24112003_01_07 24112003_01_08
24112003_01_10 24112003_01_11 24112003_01_12 24112003_01_13 24112003_01_14 24112003_01_15 24112003_01_16 24112003_01_17
24112003_01_19 24112003_01_25 24112003_02_01 24112003_02_02 24112003_02_06 24112003_02_07 24112003_02_15 24112003_02_16
24112003_02_17 24112003_02_22 24112003_02_23 24112003_03_01 24112003_03_02 24112003_03_03 24112003_03_09 24112003_03_13
24112003_03_14 24112003_03_15 24112003_04_01 24112003_04_02 24112003_04_06 24112003_04_07 24112003_04_10 24112003_04_11
24112003_04_12 24112003_04_16 24112003_04_17 24112003_04_18 24112003_05_14 24112003_05_15 24112003_05_16 24112003_05_17
24112003_05_18 24112003_05_19 24112003_05_20 24112003_05_21 24112003_05_22 24112003_05_24 24112003_26_01 24112003_26_02
24112003_26_03 24112003_26_04 24112003_26_05 24112003_26_07 24112003_04_18 22122003_06_01 22122003_06_02 22122003_06_03
22122003_06_04 22122003_06_05 22122003_06_06 22122003_06_07 22122003_06_08 22122003_06_09 22122003_06_10 22122003_06_11
22122003_06_12 22122003_06_13 22122003_06_14 22122003_06_15 22122003_06_16 22122003_06_17 22122003_06_18 22122003_06_19
22122003_06_20 22122003_06_21 22122003_06_22 22122003_06_23 22122003_06_24 22122003_06_25 22122003_07_01 22122003_07_02
22122003_07_03 22122003_07_04 22122003_07_05 22122003_07_06 22122003_07_07 22122003_07_08 22122003_08_06 22122003_08_20
22122003_08_25 22122003_10_12 22122003_10_14 22122003_10_16 22122003_11_01 22122003_11_02 22122003_11_03 22122003_11_04
22122003_11_05 22122003_11_06 22122003_11_07 22122003_11_08 22122003_11_09 22122003_11_10 22122003_11_11 22122003_11_12
22122003_11_13 22122003_11_14 22122003_16_20 22122003_17_06 22122003_17_07 22122003_17_08 22122003_17_09 22122003_26_11
22122003_26_12 23122003_17_10 23122003_17_11 23122003_17_12 23122003_17_13 23122003_17_14 23122003_17_15 23122003_17_16
23122003_17_17 23122003_17_18 23122003_17_19 23122003_17_20 23122003_17_21 23122003_18_01 23122003_18_02 23122003_18_03
23122003_18_04 23122003_18_05 23122003_18_06 23122003_18_07 23122003_18_08 23122003_18_10 23122003_18_11 23122003_18_12
23122003_18_13 23122003_18_14 23122003_18_15 23122003_18_16 23122003_18_17 23122003_18_18 23122003_18_19 23122003_18_20
23122003_18_21 23122003_18_22 23122003_18_23 23122003_18_24 23122003_18_25 23122003_19_05 23122003_19_06 23122003_19_07
23122003_19_08 23122003_19_09 23122003_19_11 23122003_19_13 23122003_19_14 23122003_19_22 23122003_19_23 23122003_19_24
23122003_19_25 23122003_20_01 23122003_20_02 23122003_20_03 23122003_20_04 23122003_20_05 23122003_20_06 23122003_20_07
23122003_20_08 23122003_20_10 23122003_20_11 23122003_20_12 23122003_20_13 23122003_20_14 23122003_20_15 23122003_20_16
23122003_20_18 23122003_20_19 23122003_20_20 23122003_20_21 23122003_20_22 23122003_20_23 23122003_20_24 23122003_20_25
23122003_21_24 23122003_22_01 23122003_22_02 23122003_22_03 23122003_22_04 23122003_22_08 23122003_22_12 23122003_22_13
23122003_22_14 23122003_22_15 23122003_22_21 23122003_22_24 23122003_22_25 23122003_23_21 23122003_24_01 23122003_24_02
23122003_24_03 23122003_24_07 23122003_24_08 23122003_24_09 23122003_25_24 23122003_25_25 23122003_26_13

Conjunto de frases para treinar os modelos:

24112003_01_01 24112003_01_02 24112003_01_03 24112003_01_04 24112003_01_05 24112003_01_06 24112003_01_07 24112003_01_08
24112003_01_10 24112003_01_11 24112003_01_12 24112003_01_13 24112003_01_14 24112003_01_15 24112003_01_16 24112003_01_17
24112003_01_19 24112003_01_25 24112003_02_01 24112003_02_02 24112003_02_06 24112003_02_09 24112003_02_12 24112003_02_14
24112003_02_15 24112003_02_16 24112003_02_17 24112003_02_20 24112003_02_22 24112003_02_23 24112003_02_24 24112003_03_01
24112003_03_02 24112003_03_03 24112003_03_06 24112003_03_08 24112003_03_09 24112003_03_13 24112003_03_14 24112003_03_15
24112003_03_16 24112003_03_19 24112003_03_22 24112003_03_24 24112003_04_01 24112003_04_02 24112003_04_03 24112003_04_05
24112003_04_06 24112003_04_07 24112003_04_10 24112003_04_11 24112003_04_12 24112003_04_13 24112003_04_16 24112003_04_17
24112003_05_02 24112003_05_05 24112003_05_08 24112003_05_09 24112003_05_10 24112003_05_12 24112003_05_14 24112003_05_15
24112003_05_16 24112003_05_17 24112003_05_18 24112003_05_19 24112003_05_20 24112003_05_21 24112003_05_22 24112003_05_24
22122003_06_01 22122003_06_02 22122003_06_03 22122003_06_04 22122003_06_05 22122003_06_06 22122003_06_07 22122003_06_08
22122003_06_09 22122003_06_11 22122003_06_12 22122003_06_13 22122003_06_14 22122003_06_15 22122003_06_16 22122003_06_18
22122003_06_19 22122003_06_20 22122003_06_21 22122003_06_22 22122003_06_23 22122003_06_24 22122003_06_25 22122003_07_01
22122003_07_02 22122003_07_03 22122003_07_04 22122003_07_05 22122003_07_07 22122003_07_08 22122003_07_09 22122003_07_10
22122003_07_11 22122003_07_14 22122003_07_15 22122003_08_02 22122003_08_05 22122003_08_07 22122003_08_08 22122003_08_10
22122003_08_19 22122003_09_05 22122003_09_07 22122003_09_16 22122003_09_19 22122003_09_23 22122003_10_03 22122003_10_09
22122003_10_12 22122003_10_13 22122003_10_14 22122003_10_15 22122003_10_19 22122003_10_20 22122003_10_21 22122003_10_22
22122003_10_23 22122003_10_24 22122003_10_25 22122003_11_01 22122003_11_02 22122003_11_03 22122003_11_05 22122003_11_06
22122003_11_07 22122003_11_08 22122003_11_09 22122003_11_11 22122003_11_12 22122003_11_13 22122003_11_14 22122003_11_15
22122003_11_16 22122003_11_17 22122003_11_18 22122003_11_19 22122003_11_20 22122003_11_21 22122003_11_22 22122003_11_23
22122003_11_24 22122003_12_01 22122003_12_02 22122003_12_05 22122003_12_06 22122003_12_07 22122003_12_08 22122003_12_09
22122003_12_12 22122003_12_13 22122003_12_14 22122003_12_16 22122003_12_18 22122003_12_19 22122003_12_20 22122003_12_22
22122003_12_23 22122003_12_25 22122003_13_01 22122003_13_04 22122003_13_05 22122003_13_06 22122003_13_07 22122003_13_08
22122003_13_09 22122003_13_10 22122003_13_12 22122003_13_13 22122003_13_14 22122003_13_15 22122003_13_16 22122003_13_17
22122003_13_19 22122003_13_20 22122003_13_21 22122003_13_22 22122003_13_23 22122003_13_25 22122003_14_01 22122003_14_03
22122003_14_04 22122003_14_06 22122003_14_07 22122003_14_08 22122003_14_10 22122003_14_12 22122003_14_13 22122003_14_14
22122003_14_15 22122003_14_16 22122003_14_17 22122003_15_09 22122003_15_10 22122003_15_23 22122003_16_02 22122003_16_05
22122003_16_16 22122003_16_18 22122003_17_05 22122003_17_06 22122003_17_07 22122003_17_08 22122003_17_09 22122003_26_10
23122003_17_10 23122003_17_11 23122003_17_12 23122003_17_13 23122003_17_14 23122003_17_15 23122003_17_16 23122003_17_17
23122003_17_18 23122003_17_19 23122003_17_20 23122003_17_21 23122003_17_22 23122003_17_24 23122003_17_25 23122003_18_01
23122003_18_02 23122003_18_03 23122003_18_04 23122003_18_05 23122003_18_06 23122003_18_07 23122003_18_08 23122003_18_11
23122003_18_12 23122003_18_13 23122003_18_14 23122003_18_15 23122003_18_16 23122003_18_17 23122003_18_18 23122003_18_19
23122003_18_20 23122003_18_21 23122003_18_22 23122003_18_24 23122003_18_25 23122003_19_01 23122003_19_03 23122003_19_04
23122003_19_05 23122003_19_06 23122003_19_07 23122003_19_08 23122003_19_09 23122003_19_11 23122003_19_12 23122003_19_13
23122003_19_14 23122003_19_16 23122003_19_17 23122003_19_18 23122003_19_19 23122003_19_21 23122003_19_22 23122003_19_23
23122003_19_24 23122003_19_25 23122003_20_01 23122003_20_02 23122003_20_03 23122003_20_04 23122003_20_05 23122003_20_07
23122003_20_08 23122003_20_10 23122003_20_11 23122003_20_12 23122003_20_13 23122003_20_14 23122003_20_15 23122003_20_16
23122003_20_18 23122003_20_19 23122003_20_20 23122003_20_21 23122003_20_22 23122003_20_23 23122003_20_24 23122003_20_25
23122003_21_02 23122003_21_03 23122003_21_04 23122003_21_06 23122003_21_07 23122003_21_08 23122003_21_09 23122003_21_10
23122003_21_11 23122003_21_13 23122003_21_15 23122003_21_17 23122003_21_18 23122003_21_19 23122003_21_20 23122003_21_21
23122003_21_23 23122003_21_24 23122003_21_25 23122003_22_01 23122003_22_02 23122003_22_03 23122003_22_04 23122003_22_05
23122003_22_06 23122003_22_07 23122003_22_08 23122003_22_10 23122003_22_12 23122003_22_13 23122003_22_14 23122003_22_15
23122003_22_19 23122003_22_20 23122003_22_21 23122003_22_22 23122003_22_23 23122003_22_24 23122003_22_25 23122003_23_01
23122003_23_03 23122003_23_04 23122003_23_05 23122003_23_07 23122003_23_08 23122003_23_10 23122003_23_11 23122003_23_12
23122003_23_14 23122003_23_15 23122003_23_16 23122003_23_18 23122003_23_19 23122003_23_22 23122003_23_23 23122003_23_24
23122003_24_01 23122003_24_02 23122003_24_03 23122003_24_04 23122003_24_05 23122003_24_06 23122003_24_07 23122003_24_08
23122003_24_09 23122003_24_17 23122003_24_19 23122003_25_07 23122003_25_13 23122003_25_20

Conjunto de frases para desenvolvimento e afinação dos modelos:

24112003_01_21 24112003_01_23 24112003_02_04 24112003_02_11 24112003_02_21 24112003_03_12 24112003_03_23 24112003_04_02
24112003_04_15 24112003_04_19 24112003_04_20 24112003_04_22 24112003_04_23 24112003_04_24 24112003_05_01 24112003_05_03
24112003_05_04 24112003_05_06 24112003_05_07 24112003_05_11 24112003_05_13 24112003_26_08 24112003_26_09 22122003_07_12
22122003_07_16 22122003_07_18 22122003_07_20 22122003_07_21 22122003_07_23 22122003_08_01 22122003_08_03 22122003_08_09
22122003_08_13 22122003_08_15 22122003_08_16 22122003_08_18 22122003_08_21 22122003_08_23 22122003_09_02 22122003_09_03
22122003_09_06 22122003_09_08 22122003_09_11 22122003_09_13 22122003_09_15 22122003_09_17 22122003_09_20 22122003_09_21
22122003_09_24 22122003_10_01 22122003_10_04 22122003_10_05 22122003_10_07 22122003_10_11 22122003_10_18 22122003_12_04
22122003_12_10 22122003_12_17 22122003_13_02 22122003_13_18 22122003_14_02 22122003_14_09 22122003_14_19 22122003_14_21
22122003_14_22 22122003_14_24 22122003_15_01 22122003_15_03 22122003_15_05 22122003_15_08 22122003_15_12 22122003_15_14
22122003_15_16 22122003_15_17 22122003_15_20 22122003_15_22 22122003_15_24 22122003_16_01 22122003_16_03 22122003_16_07
22122003_16_09 22122003_16_11 22122003_16_13 22122003_16_15 22122003_16_19 22122003_16_22 22122003_16_24 22122003_17_01
22122003_17_03 23122003_18_09 23122003_22_18 23122003_23_09 23122003_24_12 23122003_24_13 23122003_24_14 23122003_24_15
23122003_24_16 23122003_24_20 23122003_24_21 23122003_24_22 23122003_24_23 23122003_24_24 23122003_24_25 23122003_25_01
23122003_25_03 23122003_25_04 23122003_25_05 23122003_25_06 23122003_25_09 23122003_25_10 23122003_25_11 23122003_25_12
23122003_25_15 23122003_25_17 23122003_25_18 23122003_25_19 23122003_25_22 23122003_25_23

Conjunto de frases para teste final ou validação dos modelos:

24112003_01_09 24112003_01_18 24112003_01_20 24112003_01_22 24112003_01_24 24112003_02_03 24112003_02_05 24112003_02_07
24112003_02_08 24112003_02_10 24112003_02_13 24112003_02_18 24112003_02_19 24112003_02_25 24112003_03_01 24112003_03_04
24112003_03_05 24112003_03_07 24112003_03_10 24112003_03_13 24112003_03_18 24112003_03_19 24112003_03_25 24112003_04_04
24112003_04_08 24112003_04_09 24112003_04_14 24112003_04_21 24112003_04_25 22122003_07_13 22122003_07_17 22122003_07_19
22122003_07_22 22122003_07_24 22122003_08_04 22122003_08_11 22122003_08_12 22122003_08_14 22122003_08_17 22122003_08_22
22122003_08_24 22122003_09_01 22122003_09_04 22122003_09_09 22122003_09_10 22122003_09_12 22122003_09_14 22122003_09_18
22122003_09_22 22122003_09_25 22122003_10_02 22122003_10_06 22122003_10_08 22122003_10_10 22122003_10_17 22122003_11_07
22122003_11_25 22122003_12_03 22122003_12_11 22122003_12_15 22122003_12_21 22122003_12_24 22122003_13_03 22122003_13_11
22122003_13_24 22122003_14_05 22122003_14_11 22122003_14_18 22122003_14_20 22122003_14_23 22122003_14_25 22122003_15_02
22122003_15_04 22122003_15_06 22122003_15_07 22122003_15_11 22122003_15_13 22122003_15_15 22122003_15_18 22122003_15_19
22122003_15_21 22122003_15_25 22122003_16_04 22122003_16_06 22122003_16_08 22122003_16_10 22122003_16_12 22122003_16_14
22122003_16_17 22122003_16_21 22122003_16_23 22122003_16_25 22122003_17_02 22122003_17_04 23122003_17_23 23122003_18_08
23122003_19_02 23122003_19_10 23122003_19_15 23122003_19_20 23122003_21_01 23122003_21_05 23122003_21_12 23122003_21_14
23122003_21_16 23122003_21_22 23122003_22_05 23122003_22_09 23122003_22_11 23122003_22_17 23122003_23_02 23122003_23_06
23122003_23_13 23122003_23_20 23122003_23_25 23122003_24_10 23122003_24_11 23122003_24_18 23122003_25_02 23122003_25_08
23122003_25_14 23122003_25_16 23122003_25_21

ANEXO C

Tabelas de afinação do parâmetro de combinação *multi-stream*

De seguida são apresentadas as tabelas dos resultados obtidos durante a afinação do parâmetro que atribui o peso a cada um dos streams no descodificador *multi-stream*, para diferentes SNR.

Com Bigram e SNR 24 dB			
Prob	WER-Audio	WER-Video	WER-AV
0	1,12%	36,88%	36,88%
0,1	1,12%	36,88%	28,52%
0,2	1,12%	36,88%	17,04%
0,3	1,12%	36,88%	12,36%
0,4	1,12%	36,88%	8,08%
0,5	1,12%	36,88%	4,88%
0,6	1,12%	36,88%	2,76%
0,7	1,12%	36,88%	1,88%
0,8	1,12%	36,88%	1,04%
0,9	1,12%	36,88%	1,12%
1	1,12%	36,88%	1,12%

Com Bigram e SNR 24 dB			
Prob	WER-Audio	WER-Video	WER-Final
0,7	1,12%	36,88%	1,88%
0,725	1,12%	36,88%	1,36%
0,75	1,12%	36,88%	1,24%
0,775	1,12%	36,88%	1,16%
0,8	1,12%	36,88%	1,04%
0,825	1,12%	36,88%	1,00%
0,85	1,12%	36,88%	0,92%
0,875	1,12%	36,88%	0,96%
0,9	1,12%	36,88%	1,12%
0,925	1,12%	36,88%	1,08%
0,95	1,12%	36,88%	1,12%

Com Bigram e SNR 19 dB			
Prob	WER-Audio	WER-Video	WER-Final
0	2,96%	36,88%	36,88%
0,1	2,96%	36,88%	24,52%
0,2	2,96%	36,88%	17,80%
0,3	2,96%	36,88%	13,20%
0,4	2,96%	36,88%	9,08%
0,5	2,96%	36,88%	5,72%
0,6	2,96%	36,88%	3,20%
0,7	2,96%	36,88%	2,48%
0,8	2,96%	36,88%	2,76%
0,9	2,96%	36,88%	2,68%
1	2,96%	36,88%	2,96%

Com Bigram e SNR 19 dB			
Prob	WER-Audio	WER-Video	WER-Final
0,6	2,96%	36,88%	3,20%
0,625	2,96%	36,88%	3,12%
0,65	2,96%	36,88%	3,04%
0,675	2,96%	36,88%	2,88%
0,7	2,96%	36,88%	2,48%
0,725	2,96%	36,88%	2,52%
0,75	2,96%	36,88%	2,48%
0,775	2,96%	36,88%	2,40%
0,8	2,96%	36,88%	2,76%
0,85	2,96%	36,88%	2,76%
0,9	2,96%	36,88%	2,68%

Com Bigram e SNR 14 dB			
Prob	WER-Audio	WER-Video	WER-Final
0	18,24%	36,88%	36,88%
0,1	18,24%	36,88%	28,88%
0,2	18,24%	36,88%	23,12%
0,3	18,24%	36,88%	18,84%
0,4	18,24%	36,88%	15,16%
0,5	18,24%	36,88%	13,32%
0,6	18,24%	36,88%	12,44%
0,7	18,24%	36,88%	13,40%
0,8	18,24%	36,88%	16,92%
0,9	18,24%	36,88%	17,12%
1	18,24%	36,88%	18,24%

Com Bigram e SNR 14 dB			
Prob	WER-Audio	WER-Video	WER-Final
0,5	18,24%	36,88%	13,32%
0,525	18,24%	36,88%	13,32%
0,55	18,24%	36,88%	13,08%
0,575	18,24%	36,88%	12,72%
0,6	18,24%	36,88%	12,44%
0,625	18,24%	36,88%	12,40%
0,65	18,24%	36,88%	12,24%
0,675	18,24%	36,88%	12,88%
0,7	18,24%	36,88%	13,40%

Com Bigram e SNR 9 dB			
Prob	WER-Audio	WER-Video	WER-Final
0	32,52%	36,88%	36,88%
0,1	32,52%	36,88%	33,24%
0,2	32,52%	36,88%	32,04%
0,3	32,52%	36,88%	31,60%
0,4	32,52%	36,88%	29,12%
0,5	32,52%	36,88%	26,64%
0,6	32,52%	36,88%	24,08%
0,7	32,52%	36,88%	28,44%
0,8	32,52%	36,88%	27,68%
0,9	32,52%	36,88%	32,04%
1	32,52%	36,88%	32,52%

Sem Bigram-Wordpair e SNR 9 dB			
Prob	WER-Audio	WER-Video	WER-Final
0,5	32,52%	36,88%	26,64%
0,525	32,52%	36,88%	25,16%
0,55	32,52%	36,88%	24,04%
0,575	32,52%	36,88%	24,08%
0,6	32,52%	36,88%	24,08%
0,625	32,52%	36,88%	25,52%
0,65	32,52%	36,88%	27,12%
0,675	32,52%	36,88%	28,04%
0,7	32,52%	36,88%	28,44%

Com Bigram e SNR-20 dB			
Prob	WER-Audio	WER-Video	WER-Final
0	50,16%	36,88%	36,88%
0,1	50,16%	36,88%	36,80%
0,2	50,16%	36,88%	36,40%
0,3	50,16%	36,88%	35,08%
0,4	50,16%	36,88%	33,56%
0,5	50,16%	36,88%	33,56%
0,6	50,16%	36,88%	38,44%
0,7	50,16%	36,88%	44,92%
0,8	50,16%	36,88%	43,56%
0,9	50,16%	36,88%	46,68%
1	50,16%	36,88%	50,16%

Sem Bigram-Wordpair e SNR-20 dB			
Prob	WER-Audio	WER-Video	WER-Final
0,3	50,16%	36,88%	35,08%
0,325	50,16%	36,88%	34,92%
0,35	50,16%	36,88%	34,12%
0,375	50,16%	36,88%	33,80%
0,4	50,16%	36,88%	33,56%
0,425	50,16%	36,88%	33,88%
0,45	50,16%	36,88%	34,04%
0,475	50,16%	36,88%	34,72%
0,5	50,16%	36,88%	33,56%
0,525	50,16%	36,88%	34,76%
0,55	50,16%	36,88%	36,08%

BIBLIOGRAFIA

[1] Rabiner, L. R., Juang, B-H.: Fundamentals of Speech Recognition. Prentice Hall, 1993.

[2] Deller, J. R. Jr., Proakis, J. G., Hansen, J. H. L.: Discrete-Time Processing of Speech Signals. Macmillan, 1993.

[3] Proakis, John G., Deller, John R., Hansen, John H. L.: Discrete-Time Processing of Speech Signals. IEEE Press, 2000.

[4] Parsons, T.: Voice And Speech Processing. Mc Graw-Hill, 1987.

[5] Ribeiro, Carlos: Processamento Digital de Fala. Abril 2003.

[6] Cintra, Lindley, Cunha, Celso: Nova Gramática do Português Contemporâneo. Edições João Sá da Costa, Lda, 16a edition, 2000.

[7] Martins, M.: Ouvir Falar - Introdução à Fonética do Português. Ed. Caminho, Lisboa (Portugal), 2a. Edição, 1988.

[8] Prof. MUDr. DrSc. Cihak, R.: Anatomie 1, pages 361-380, Prague 1987.

[9] Sinelnikov, R.D.: Atlas anatomie cloveka 1, pages 284-297, Moscow 1978.

[10] Fisher, C.: Confusions among visually perceived consonants. Journal of Speech and Hearing Research, vol. 11, pp. 796-804, 1968.

[11] Jeffers, J., Barley, M.: Speechreading (Lipreading). Charles C. Thomas Publisher, 1971.

[12] Pickett, J.: The sounds of speech communication. Baltimore, MD: University Park Press, 1980.

[13] Liberman, A., Cooper, D., Shankweiler, F. S., Studdert-Kennedy, M.: Perception of the speech code. Psychological Review, vol. 25, pp. 600-607, 1982.

[14] Benguerel, A., Pichora-Fuller, M.: Coarticulation effects in lipereading. Journal oF Speech and Hearing Research, vol. 25, pp 600-607, 1982.

[15] Montgomery, A.: Development of a model for generating synthetic animated lip shapes. JASA, p. S58, 1980.

[16] Williams, J.: Speech-to-Video Coversion for Individuals With Ipaired Hearing. PhD thesis,, Northwestern University, 2000.

[17] Jackson, P.: The theoretical minimal unit for visual speech perception: Visemes and coarticulation. The Volta Review, vol. 90, Nº 5, pp 99-115, 1988.

[18] Pêra, V.: Reconhecimento Automático de Fala Contínua Com Processamento Simultâneo de Diferentes Características do Sinal, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 2001.

[19] Rodrigues, F.: Reconhecimento Robusto de Dígitos e Números Naturais. MSc thesis, Instituto Superior Técnico da Universidade Técnica de Lisboa, 2001.

[20] Junqua, J. C.: The Influence of Acoustics on Speech Production: a Noise-Induced Stress Phenomenon Known as the Lombard Reflex. ESCA-NATO Tutorial and Research Workshop on "Speech Under Stress", Lisboa, Portugal, Setembro de 1995.

[21] Paviwal, K. K.: Robust Speech Recognition, Griffith University.

[22] Junqua, J.-C., Haton, J.-P., Robustness in Automatic Speech Recognition – Fundamentals and Applications, Kluwer Academic Publishers, 1996.

[23] Hanson, B. A., Applebaum, T. H., Robust speaker-independent word recognition using static, dynamic and acceleration features. IEEE International Conference on Acoustics, Speech and Signal Processing. Pp. 857-860, 1990

[24] Haton, J. P.:Automatic Recognition of Noisy Speech. Speech Recognitionand Coding - New Advances and Trends, NATO Advanced Science Institute Series, Editado por A. R. Ayuso e J. L. Soler, Springer-Verlag, págs. 3-14, 1995.

[25] Lee, C. H.: On Stochastic Feature and Model Compensation Approaches to Robust Speech Recognition. Speech Communication, nº 25, págs. 29-47, 1998.

[26] Furui, S.: Robust Speech Recognition. NATO Advanced Science InstituteSeries: Computational Models of Speech Pattern Processing, Editado por K. Ponting, Springer-Verlag, págs. 102-111, 1997.

[27] Project-Team Parole Analysis, Perception and speech recognition Lorraine, INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE, 2004.

[28] Hermansky, H.: Perceptual Linear Predictive (PLP) Analysis ofSpeech. Journal of Acoustic Society of America, Abril de 1990.

[29] Perdigão, F.: Modelos do Sistema Auditivo Periférico no Reconhecimento Automático de Fala. Tese de Doutoramento, Universidade de Coimbra, Coimbra, Outubro de 1997.

[30] Knill, K., Young, S.: Hidden Markov Models in Speech and Language Processing. Corpus-Based Methods in Language and Speech Processing, Editado por S. Young e G. Bloothooft, Kluwer Academic Publishers, 1997.

[31] Boll, S. F.: Supression of Acoustic Noise in Speech Using Spectral Subtraction. IEEE Transactions On Acoustics, Speech and Signal Processing, Abril de 1979.

[32] Atal, B.: Effectiveness of linear prediction characteristics of the speech wave forautomatic speaker identification and verification. Journal of the Acoustical Society of America, 55:1304–1312, June 1974.

[33] Furui, S.: Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-29(2):254–272, September 1996.

[34] Hermansky, H., Morgan, N.: Rasta Processing of Speech. IEEE Transactions On Speech and Audio Processing, Vol. 2, nº 4, Outubro de 1996.

[35] Rahim, M. G., Juang, B. H.: Signal Bias Removal by Maximum Likelihood Estimation for Robust Telephone Speech Recognition. IEEE Transactions On Speech and Audio Processing, Vol. 4, nº 1, págs. 19-30, Janeiro de 1996.

[36] Gales, M. J. F., Young, S. J.: Robust Speech Recognition in Additive and Convolutional Noise Using Parallel Model Combination. Computer Speech and Language, Vol. 9, 1995.

[37] Gales, M., Young, S.: An improved approach to the hidden Markov model decomposition of speech and noise. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:233–236, March 1992.

[38] Gales, M., Young, S.: Robust continuous speech recognition using parallel model combination. IEEE Transactions on Speech and Audio Processing, 4(5):352–359, September 1996.

[39] Junqua, J. C.: ICASSP'99 Tutorial – Robust Automatic Speech Recognition for Unknown Environment Compensation. Phoenix, EUA, Março de 1999.

[40] Sankar, A., Lee, C. H.: Robust Speech Recognition Based on Stochastic Matching. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95), Detroit, EUA, Abril de 1995.

[41] Sankar, A., Lee, C.: A maximum likelihood approach to stochastic matching for robust speech recognition. IEEE Transactions on Speech and Audio Processing, 4(3):190–202, May 1996.

[42] Ortega-Garcia, J., Gonzáles-Rodriguez, J.: Overview of Speech Enhacement Techniques for Automatic Speaker Recognition.

[43] Widrow, B.: Stearns, S. D.: Adaptative Signal Processing. Pretince-Hall, 1985.

[44] Duchateau, J., Laureys, T., Wambacq, P.: Adding Robustness to Language Models for Spontaneous Speech recognition. Katholieke Universiteit Leuven.

[45] Gold, B., Morgan, N., Speech and audio signal processing – processing and perception of speech and music, John Wiley & Sons, 2000.

[46] Golomb, J., Speech Recognition History
http://florin.stanford.edu/~t361/Fall2000/jgolomb/YYSpeechHistoryWB.htm, 2003

[47] Jurafsky, D., Martin, J. H.: Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice-Hall, 2000.

[48] Dixon, N. Rex and Martin, Thomas B, ed. Automatic Speech & Speaker Recognition. New York: IEEE Press, 1978.

[49] Widrow, B.: Personal Comunication, Phoenix, Az., 1999.

[50] Junqua, J.-C., Robustness and Cooperative Multimodal Man-Machine Communication Applications, Second Venaco Workshop: The Structure of Multimodal Dialogue, 1991.

[51] Petajan, E.D.: Automatic lipreading to enhance speech recognition. Proc. Global Telecommunications Conference, Atlanta, GA, pp. 265–272, 1984.

[52] Potamianos, G., Neti, C., Gravier, G:, Garg, G., Senior, A. W., "Recent Advances in Automatic Recognition of Audio-Visual Speech", Procedins of the IEEE, vol. 91, nº9, September, 2003.

[53] Hennecke, M. E., Stork, D. G., Prasand, K. V.: Visionary speech: Looking ahead to pratical speechreading systems. Speechreading by Humans and machines, D. G. Stork and M. E. Hennecke, Eds. Berlin, Germany: Springer pp. 331–349, 1996.

[54] Su, Q. and Silsbee, P.L. Robust audiovisual integration using semicontinuous hidden Markov models. Proc. International Conference on Spoken Language Processing, Philadelphia, PA, pp. 42–45, 1996.

[55] Adjoudani, A., Benoît, C.: On the integration of auditory and visual parameters in an HMM-based ASR. In Stork, D.G. and Hennecke, M.E. (Eds.), Speechreading by Humans and Machines. Berlin, Germany: Springer, pp. 461–471, 1996.

[56] Potamianos, G., Graf, H.P., and Cosatto, E.: An image transform approach for HMM based automatic lipreading. Proc. International Conference on Image Processing, Chicago, IL, vol. I, pp. 173–177, 1998.

[57] Zhang, Y., Levinson, S., and Huang, T. Speaker independent audio-visual speech recognition. Proc. International Conference on Multimedia and Expo, New York, NY, pp. 1073–1076, 2000.

[58] Goldschen, A.J., Garcia, O.N., and Petajan, E.D.: Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. In Stork, D.G. and Hennecke, M.E. (Eds.), Speechreading by Humans and Machines. Berlin, Germany: Springer, pp. 505–515, 1996.

[59] Rogozan, A., Deléglise, P., and Alissali, M.: Adaptive determination of audio and visual weights for automatic speech recognition. Proc. European Tutorial Workshop on Audio-Visual Speech Processing, Rhodes, Greece, pp. 61–64, 1997.

[60] Bregler, C., Konig, Y. (1994). " Eigenlips" for robust speech recognition. Proc. International Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia, pp. 669–672.

[61] Nakamura, S., Ito, H., and Shikano, K.: Stream weight optimization of speech and lip image sequence for audiovisual speech recognition. Proc. International Conference on Spoken Language Processing, Beijing, China, vol. III, pp. 20–23, 2000.

[62] Huang, F.J., Chen, T.: Consideration of Lombard effect for speechreading. Proc. Workshop on Multimedia Signal Processing, Cannes, France, pp. 613–618, 2001.

[63] Martens, J. P., Final Report of COST Action 249 – Introduction, European Commission, 2000.

[64] El-Koura, K., Speech Recognition – a web site by Karl El-Koura, http://nexus.ca/~kekoura/index.html, 2003

[65] Raposo, Eduardo Paiva, Teoria da Gramática. A Faculdade da Linguagem Editorial Caminho, 2ª edição, pp. 65-85, Setembro de 1998.

[66] Neto, J. P., Martins, C., Almeida, L.: A large vocabulary continuous speech recognition for The Portuguese Language. Instituto Superior Técnico, 1998.

[67] Pera, V., S, F., Afonso, P., and Ferreira, R.: Audio-Visual Speech Recognition in a Portuguese Language Based Application. Proceedings of the International Conference on Industrial Technology, Maribor, Slovenia, 2003.

[68] Bregler, C., Hild, H., Manke, S., and Waibel, A.: Improving connected letter recognition by lipreading. Proc. International Conference on Acoustics, Speech and Signal Processing, Minneapolis, MN, pp. 557–560, 1993

[69] Potamianos, G., Graf, H.P.: Discriminative training of HMM stream exponents for audio-visual speech recognition. Proc. International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, pp. 3733–3736, 1998.

[70] Rogozan, A. Discriminative learning of visual data for audiovisual speech recognition. International Journal on Artificial Intelligence Tools, 8(1):43–52, 1999.

[71] Teissier, P., Robert-Ribes, J., and Schwartz, J.L.: Comparing models for audiovisual fusion in a noisy-vowel recognition task. IEEE Transactions on Speech and Audio Processing 7(6):629–642, 1999.

[72] Dupont, S. and Luettin, J., "Audio-visual speech recognition for continuous speech recognition," IEEE Trans. Multimedia, vol. 2, pp. 141-151, 2000.

[73] Chen, T.: Audio-visual speech processing. Lip reading and lip synchronization. IEEE Signal Processing Magazine, 18(1):9–21, 2001.

[74] Chu, S., Huang, T.: Audio-visual speech modelling using coupled hidden Markov models. Proc. International Conference on Acoustics, Speech and Signal Processing, Orlando, FL, pp. 2009–2012, 2002.

[75] Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., Vergyri, D., Sison, J., Mashari, A., and Zhou, J.: Audio-Visual Speech Recognition. Final Workshop 2000 Report. Baltimore, MD: Center for Language and Speech Processing, The Johns Hopkins University, 2000.

[76] Potamianos, G., Luettin, J., and Neti, C.: Hierarchical discriminant features for audio-visual LVCSR. Proc. International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, UT, pp. 165–168, 2001a.

[77] Girin, L., Feng, G., and Schwartz, J.-L.: Noisy speech enhancement with filters estimated from the speaker's lips. Proc. European Conference on Speech Communication and Technology, Madrid, Spain, pp. 1559–1562, 1995.

[78] Baker, J. K., "The dragon system - An overview", IEEE Trans. ASSP, vol ASSP-23, n° 1 pp-24-29, Fevereiro de 1975.

[79] Grant, K.W. and Greenberg, S.: Speech intelligibility derived from asynchronous processing of auditory-visual information. Proc. International Conference on Auditory-Visual Speech Processing, Aalborg, Denmark, pp. 132–137, 2001.

[80] Stork, D. G., Hennecke, M. E.: Speechreading by Humans and Machines Models Systems and Applications. Computer and Systems Sciences. SpringerVerlag Nova Iorque.

[81] Gray, M.S., Movellan, J.R., Sejnowski, T.J.: Dynamic features for visual speech-reading: A systematic comparison In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, vol. 9, pp. 751–757, 1997.

[82] Koenig, W.: A New Frequency Scale for Acoustic Measurements. Bell Telephone Laboratory Record, 1949.

[83] Milner, B.: Inclusion of temporal information into features for speech recognition.

[84] Kumar, N, Andreou, A. G.: Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. Speech Comunication, vol. 26, pp. 283-297, 1998.

[85] Stiefelhagen R., Yang J., and Meier U.: Real-time lip tracking for lip reading. Proceedings of Eurospeech 97, 1997.

[86] Basu, S., Neti, C., Rajput N., Senior, A., Subramaniam, L., Verma, A.,: Audio-Visual Large Vocabulary Continuous Speech Recognition in the Broadcast Domain. Proceedings of Multimedia Signal Processing, Copenhagen, 1999.

[87] Kass M., Witkin A., and Terzopoulos D.: Snakes: active contour models, International Journal of Computer Vision: vol 1 pp. 321-332, 1998.

[88] Aubert, G., Barlaud, M., Faugeras O., Jehan-Besson, S.: Image Segmentation using Active Contours: Calculous of Variations of or Shape Gradients?. 2002.

[89] Hennecke, M.E, Prasad K.V., Stork, D.G.: Using deformable templates to infer visual speech dynamics. 28th Annual Asimolar Conference on Signals, Systems, and Computer: vol 2 pp. 576-582, Pacific Grove, CA. IEEE Computer, 1994.

[90] Benoit, C., Martin, J-C., Pelachaud, C., Schomaker, L., Suhm, B.: Audiovisual and Multimodal Speech Systems.

[91] Lewis T. and Powers D.: Lip Feature Extraction Extraction using Red Exclusion". World Wide Web, www.cs.usyd.edu.au/~vip2000 , 2000.

[92] Matthews, I., Potamianos, G., Neti, C., Luettin, J.: A comparison of model and transform-based visual features for audio-visual LVCSR. (In Press), Proc. ICME, 2001.

[93] Chan, M.: HMM-Based Audio-Visual Speech Recognition Integrating Geometric and Appearance-Based Visual Features. Proceedings of IEEE Workshop on Multimedia Signal Processing, pp. 9-14, Cannes, France, 2001.

[94] Chan, M.: Automatic Lip Model Extraction for Constrained Contour-Based Tracking. Proceedings of the IEEE International Conference on Image Processing, Vol. 2, 848-851, Kobe, 1999.

[95] Baum, L. E., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. Vol. 37, pp. 1554-1563, 1966.

[96] Jelinek, F.: A fast sequential decoding algorithm using a stack. IBM J. Res. Develop., vol 13, pp. 675-685, 1969.

[97] Jelinek, F., Bahl, L. R., Mercer, R. L.: Design of a linguistic statistical decoder for the recognition of continuos speech. IEEE Trans. Informat. Theory, vol IT-21, pp.250-256, 1975.

[98] Bahl, L. R., Jelinek, F.:Decoding for channels with insertions deletions and substitutions with applications to speech recognition. IEEE Trans. Informat. Theory, vol. IT-21, pp404-411, 1975.

[99] Juang, B. H.: On the Hidden Markov Model and dynamic time warping for speech recognition - A unified view. AT&T Tech. J. vol 63. N°7, pp. 1213-1243, September 1984.

[100] Levinson, S. E., Rabiner, L. R., Shondi, M. M.: An introduction to the application of the theory of probabilistic function of a Markov process to automatic speech recognition., Bell Syst. Tech. J. vol 62, n°4, pp. 1035-1074, Abril de 1983.

[101] Rabiner, L. R.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the IEEE, Vol 77, n°2, Fevereiro de 1989.

[102] Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Ed. Prentice-Hall, New Jersey (EUA), 1993.

[103] Huang, X. D., Hwang, M. Y., Lee, K. F.: Hidden Markov models for speech recognition. Computer Speech and Language, vol.(3), pp.239-252, 1989.

[104] Huang, X. D., Jack, M. A.: Semi-continuous hidden Markov models for speech recognition. Computer Speech and Language, vol.(3), pp.239-252, 1989.

[105] Lee, K. F., Alleva, F.: Continuous Speech Reconition_

[106] Strom, N.: A tonotopic artificial neural network architecture for phoneme probability estimation. In Proc. ASRU ’97, 1997.

[107] Peinado, A. M., Segura, J. C., Rubio, A. J., Garcia, P, Pérez, J. L.: Discriminative codebook design using multiple vector quantization in HMM-based speech

recognizers. IEEE Transactions on Speech and Audio Processing, vol. (4), nº2, Março 1996.

[108] Dermatas, E., Kokkinakis, G.: Algorithm for clustering continuous density HMM by recognition error. IEEE Transactions on Speech and Audio processing, vol.(4), nº3, Maio 1996.

[109] Imperl, B., Kohler, J., Kacic, Z.: On the use of semi-continuous HMM for the isolated digits recognition over the telephone.

[110] Bourland, H., Dupont, S., Ris, C.: Multi-stream speech recognition. IDIAP Research Report, Mons, 1996..

[111] Miller, L., Levinson, S.: Syntactic Analysis for Large Vocabulary Speech Recognition using a Context-Free Covering Grammar. In Proceedings ICASSP 88, pp. 270–274, 1988.

[112] Pieraccini, R., Lee, C., Giachin, E., Rabiner, L.: Complexity reduction in a Large Vocabulary Speech Recognizer. In Proceedings ICASSP 91, pp. 729–732, 1991.

[113] Markowitz, J.: Using Speech Recognition. Ed. Prentice-Hall, New Jersey, EUA, 1996.

[114] Clarkson, P.R., Rosenfeld, R.: Statistical Language Modeling Using the CMU-Cambridge Toolkit. Proceedings of the Eurospeech, Rhodes, Greece, 1997.

[115] Paterson, E. K.: Audio Visual Speech Recognition for Difficult Environments. PhD thesis, Clemson University, 2002.

[116] Weber, K., Ikbal, S., Bengio, S., Bourlard, H.: Robust Speech Recognition and Feature Extraction Using HMM2. Computer Speech & Language, 17, 2003.

[117] Bailly-Baillire, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Marithoz, J., Matas, J., Messer, K., Popovici, V., Pore, F., Ruiz, B., and Thiran, J.-P.: The BANCA Database and evaluation protocol. 4th International Conference on Audio-and Video-Based Biometric Person Authentication, 2003.

[118] Warren, P.: NZSED: Building and using a speech database for New Zealand English. New Zealand Journal, 1(6), 2002.

[119] Chibelushi, C.C., Deravi, F., Mason, J.S.D.: Survey of Audio Visual Speech Databases. Technical Report. Swansea, United Kingdom: Department of Electrical and Electronic Engineering, University of Wales, 1996.

[120] Chibelushi, C.C., Deravi, F., and Mason, J.S.D.: A review of speech-based bimodal recognition. IEEE Transactions on Multimedia, 4(1):23–37, 2002.

[121] Robert-Ribes, J., Piquemal, M., Schwartz, J.-L., Escudier, P.: Exploiting sensor fusion architectures and stimuli complementarity in AV speech recognition. In Stork, D.G. and Hennecke, M.E. (Eds.), Speechreading by Humans and Machines. Berlin, Germany: Springer, pp. 193–210, 1996.

[122] Movellan, J.R., Chadderdon, G. Channel separability in the audio visual integration of speech: A Bayesian approach. In Stork, D.G. and Hennecke, M.E. (Eds.), Speechreading by Humans and Machines. Berlin, Germany: Springer, pp. 473–487, 1996.

[123] Luettin, J., Thacker, N.A., Beet, S.W.: Speechreading using shape and intensity information. Proc. International Conference on Spoken Language Processing, Philadelphia, PA, pp. 58–61, 1996.

[124] Vanegas, O., Tanaka, A., Tokuda, K., Kitamura, T.: HMM-based visual speech recognition using intensity and location normalization. Proc. International Conference on Spoken Language Processing, Sydney, Australia,pp. 289–292, 1998.

[125] Scanlon, P. and Reilly, R.: Feature analysis for automatic speechreading. Proc. Workshop on Multimedia Signal Processing, Cannes, France, pp. 625–630, 2001.

[126] Pigeon, S., Vandendorpe, L.: The M2VTS multimodal face database. In Bigun, J., Chollet, G., and Borgefors, G. (Eds.), Audio-and Video-based Biometric Person Authentication, Berlin, Germany: Springer, pp. 403–409, 1997.

[127] Miyajima, C., Tokuda, K., Kitamura, T.: Audio-visual speech recognition using MCE-based HMMs and model-dependent stream weights. Proc. International Conference on Spoken Language Processing, Beijing, China, vol. II, pp. 1023–1026, 2000.

[128] Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: XM2VTS: The extended M2VTS database. Proc. International Conference on Audio and Video-based Biometric Person Authentication, Washington, DC, pp. 72–76, 1999.

[129] Tomlinson, M.J., Russell, M.J., Brooke, N.M. Integrating audio and visual information to provide highly robust speech recognition. Proc. International Conference on Acoustics, Speech and Signal Processing, Atlanta, GA, pp. 821–824, 1996.

[130] Chu, S. and Huang, T.: Bimodal speech recognition using coupled hidden Markov models. Proc. International Conference on Spoken Language Processing, Beijing, China, vol. II, pp. 747–750, 2000.

[131] Meier, U., Hurst, W., Duchnowski, P.: Adaptive bimodal sensor fusion for automatic speechreading. Proc. International Conference on Acoustics, Speech and Signal Processing, Atlanta, GA, pp. 833–836, 1996.

[132] Krone, G., Talle, B., Wichert, A., Palm, G.: Neural architectures for sensorfusion in speech recognition. Proc. European Tutorial Workshop on Audio-Visual Speech Processing, Rhodes, Greece, pp. 57–60, 1997.

[133] Alissali, M., Del´eglise, P., Rogozan: AAsynchronous integration of visual information in an automatic speech recognition system. Proc. International Conference on Spoken Language Processing, Philadelphia, PA, pp. 34–37, 1996.

[134] André-Obrecht, R., Jacob, B., Parlangeau, N.: Audio visual speech recognition and segmental master slave HMM. Proc. European Tutorial Workshop on Audio-Visual Speech Processing, Rhodes, Greece, pp. 49–52, 1997.

[135] Jourlin, P.: Word dependent acoustic-labial weights in HMM-based speech recognition. Proc. European Tutorial Workshop on Audio-Visual Speech Processing, Rhodes, Greece, pp. 69–72, 1997.

[136] Matthews, I., Bangham, J.A., and Cox, S.: Audio-visual speech recognition using multiscale nonlinear image decomposition. Proc. International Conference on Spoken Language Processing, Philadelphia, PA, pp. 38–41., 1996.

[137] Cox, S., Matthews, I., Bangham, A.: Combining noise compensation with visual information in speech recognition.Proc. European Tutorial Workshop on Audio-Visual Speech Processing, Rhodes, Greece, pp. 53–56, 1997.

[138] Silsbee, P.L., Bovik, A.C.: Computer lipreading for improved accuracy in automatic speech recognition. IEEE Transactions on Speech and Audio Processing, 4(5):337–351, 1996.

[139] Chiou, G., Hwang, J.-N.: Lipreading from color video. IEEE Transactions on Image Processing, 6(8):1192–1195, 1997.

[140] Gurbuz, S., Tufekci, Z., Patterson, E., Gowdy, J.N.: Application of affi ne-invariant Fourier descriptors to lipreading for audio-visual speech recognition. Proc. International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, UT, pp. 177–180, 2001.

[141] Patterson, E.K., Gurbuz, S., Tufekci, Z., Gowdy, J.N. Noise-based audio-visual fusion for robust speech recognition. Proc. International Conference on Auditory-Visual Speech Processing, Aalborg, Denmark, pp.196–199, 2001.

[142] Kober, R., Harz, U., Schiffers: J. Fusion of visual and acoustic signals for command-word recognition. Proc. International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 1495–1497, 1997.

[143] Nakamura, S.: Fusion of audio-visual information for integrated speech processing. In Bigun, J. and Smeraldi, F. (Eds.), Audio-and Video-Based Biometric Person Authentication. Berlin, Germany: Springer-Verlag, pp.127–143, 2001.

[144] Chan, M.T., Zhang, Y., Huang, T.S.: Real-time lip tracking and bimodal continuous speech recognition. Proc. Workshop on Multimedia Signal Processing, Redondo Beach, CA, pp. 65–70, 1998.

[145] Polymenakos, L., Olsen, P., Kanevsky, D., Gopinath, R.A., Gopalakrishnan, P.S., and Chen, S.: Transcription of broadcast news - some recent improvements to IBM's LVCSR system. Proc. International Conference on Acoustics, Speech, and Signal Processing, Seattle, WA, pp. 901–904, 1998.

[146] Moura, A., Pera, V., Freitas, D.: "A New Multi-modal Database for Developing Speech Recognition Systems for na Assistive technology Application. TSD 2004, Brno, Rep. Checa, Springer-Verlag, pp. 385-392, 2004"

[147] Chen, S. F.: Bayesian Gramma Induction for Language Modelling, 1995.

[148] Neto, J. P.: Reconhecimento da Fala Contínua com aplicação de técnicas de Adaptação ao Orador. PhD Thesis, IST, 1998.

[149] Chen, S. F., Goodman, J.: An Empirical Study of Smoothing Techniques for Language Modelling. Proc. of ACL1996.

[150] Goodman, J.: A Bit of Progress in Language Modelling, Extended Version. 2001.

[151] Peng, F., Schuurmans, D.: Combining Naive Bayes and n-Gram Language Models for Text Classication. 25th European Conference on IR Research, Pisa, Italy, 2003.

[152] Witten, I.H., Bell, T. C.: The Zero-Frequency Problem: Estimating the Probabilities of novel Events in Adaptive Text Compression. In IEEE Transactions on Information Theory, vol. 37, Nº4, 1991.

[153] Efron, A.: The jacknife, the bootstrap and other resampling plans. Regional Conference Series in Applied Mathematics, Philadelphia, U.S.A., 1982.

[154] Patterson, E.K., Gurbuz, S., Tufekci, Z., and Gowdy, J.N.: CUAVE: A new audio-visual database for multimodal human-computer interface research. Proc. International Conference on Acoustics, Speech and Signal Processing, Orlando, FL, pp. 2017–2020, 2002.

[155] Lima, C.: Speech recognition in non-stationary environments. Dissertação: para Doutor em Engenharia Electrónica Industrial, Universidade do Minho, 2000.

[156] Teixeira, C.: Reconehcimento de Fala de Oradores Estrangeiros. Tese de Doutoramento, Instituto Superior Técnico, Lisboa, Setembro de 1998;

[157] Brooke, N.M.: Using visual component in automatic speech recognition. Proc. International Conference on Spoken,Language Processing, Philadelphia, USA, 1996.