Orlando Oliveira Costa

DESENVOLVIMENTO DE TÉCNICAS PARA AVALIAÇÃO AUTOMÁTICA DA CAPACIDADE DE LEITURA DAS CRIANÇAS

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores

Setembro de 2015

UNIVERSIDADE DE COIMBRA

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

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores

Desenvolvimento de Técnicas para Avaliação Automática da Capacidade de Leitura das Crianças

Orlando Oliveira Costa

Desenvolvido com a Supervisão de
Professor Doutor Fernando Santos Perdigão
e coorientada por Jorge Proença

Júri:
Professor Doutor Manuel Marques Crisóstomo (Presidente)
Professor Doutor Fernando Santos Perdigão (Orientador)
Professora Doutora Rita Cristina Girão Coelho da Silva (Vogal)
Professora Adjunta Carla Alexandra Calado Lopes (Vogal)

Setembro de 2015

Agradecimentos

Agradeço de um modo especial ao Prof. Doutor Fernando Perdigão pela oportunidade, ajuda e motivação na realização desta dissertação e por todos os conhecimentos transmitidos. Agradeço também a todos os elementos do Laboratório de Processamento de Sinal, em especial ao Jorge Proença pela grande ajuda e tempo despendido no desenvolvimento deste trabalho.

Um profundo agradecimento à minha família por todo o seu esforço e apoio ao longo do meu percurso académico e por todo o seu amor, paciência e dedicação ao longo da minha vida.

Um agradecimento especial à minha namorada, Raquel Santos, por todo o apoio, amor e paciência durante este período de menor disponibilidade.

Por último e não menos importante, aos meus colegas e amigos que me deram apoio neste percurso académico inesquecível, que levo comigo para a vida. A eles, um grande obrigado!

Resumo

Com a entrada em vigor das metas curriculares de Português do 1º ciclo do ensino básico, a avaliação da capacidade de leitura das crianças exige um esforço enorme por parte dos professores. A tecnologia do reconhecimento automático da fala pode dar uma ajuda valiosa neste contexto, analisando o sinal de fala enquanto as crianças estão a ler e detetando disfluências e erros de leitura.

Esta dissertação contribuiu para o objetivo da avaliação automática da capacidade de leitura das crianças, criando modelos acústicos para o reconhecimento automático da fala e propondo técnicas de deteção de erros de pronunciação.

Na criação dos modelos recorreu-se ao treino de modelos acústicos baseados em modelos de Markov não observáveis com contexto fonético, utilizando uma base de dados com fala de crianças dos 6 aos 10 anos de idade. A adaptação de modelos foi uma alternativa estudada ao treino anteriormente referido.

Foi proposto um método de deteção de erros de pronunciação baseado no método de testes de hipóteses e na descodificação de *Viterbi*, que permite detetar as ocorrências de palavras-chaves, e através da análise das probabilidades *á posteriori* dos fones, permite identificar ocorrências de fones erradas.

Palavras-Chave: HMM, HTK, Treino de modelos acústicos, Fala de crianças, Deteção de erros de pronunciação, Reconhecimento de fala

Abstract

With the entry of curricular goals of the 1st year of primary school, the evaluation of reading skills children require a lot of effort from the teacher. The automatic speech recognition technology can give a valuable help in this task, analyzing the speech sign while the children are reading and detecting disfluencies and mistakes in reading.

This thesis contributes for the goals of automatic evaluation of reading skills of children, creating acoustic models for automatic recognition of speech and propose techniques to detect pronunciation mistakes.

For the models creation, we used the training of acoustic models based on Hidden Markov Models with phonetic context, using a data base with speech of children of 6 to 10 years old. The adaptation of the models was an alternative studied for the training previously mentioned.

A method was proposed for the detection of pronunciation mistakes based on the hypothesis testing method and on Viterbi decodification that allow to detect the keyword events, and by the analysis of the *á posteriori* probabilities of phones, allow to identify the wrong phones occurrences.

Keywords: HMM, HTK, Acoustic model training, Speech children, Detection of pronunciation errors, Speech Recognition

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

HMM – Hidden-Markov Model (modelos de Markov não observáveis)

HTK – Hidden Markov Model Toolkit

LLR – Log-Likelihood Ratio

LR – Likelihood Ratio

MAP – Maximum A Posteriori

CNG – Contents for Next Generation Networks

DET – Detection Error Tradeoff

MFCC – Mel Frequency Cepstral Coefficients

FRR– False Reject Ratio

FAR – False Accept Ratio

MLP – Multilayer Perceptron

ANN - Artificial Neural Network

PDF - Probability Distribution Function

DFT - Discrete Fourier Transform

DCT - Discrete Cossine Transform

Capítulo 1

Introdução

A leitura é uma atividade complexa que requer coordenação simultânea de várias tarefas [1]. Considera-se uma leitura fluente quando esta tem um desempenho acelerado, sem esforço e é alcançado sem muita consciência. Não sendo uma medida de compreensão é considerada um indicador da competência da leitura [2]. A avaliação do desempenho da leitura de uma criança, entre os 6 e os 10 anos, em termos de fluência envolve avaliar a velocidade, a precisão e a expressão da leitura [3], atividade que requer uma grande esforço e disponibilidade por parte do avaliador. Com o intuito de facilitar e auxiliar a tarefa do avaliador, a tecnologia do reconhecimento automático de fala tem ganho uma grande importância, nomeadamente na área de deteção de erros e disfluências na leitura.

Os principais eventos linguísticos que afetam a fluidez do discurso são as difluências como hesitações, repetições, erros de pronúncia, e tentativas de correção. A deteção automática destas disfluências é indispensável no desenvolvimento de um sistema de avaliação automático da fluência da leitura e requer o treino de um sistema de reconhecimento automático de fala. Na deteção de disfluências são usadas pesquisas de estruturas gramaticais especializadas a nível fonético [6,7,8] ou gramáticas livres [9]. Todos estes métodos de deteção são baseados nos modelos de Markov não observáveis (Hidden Markov Models - HMM), ou modelos mais complexos (como Maximum Entropy, Conditional Random Fields and Classification [4] e Regression Trees [5]).

1.1 Motivação

Esta dissertação decorre no âmbito do projeto LetsRead [10] que pretende desenvolver técnicas automáticas para avaliar a capacidade de leitura de crianças do 1º ciclo de escolaridade, tendo como motivação as metas curriculares propostas por [11], que pretende implementar uma avaliação mais objetiva do desempenho na leitura das crianças. O principal problema a resolver consiste na deteção de erros na pronunciação de palavras, pois requer modelos acústicos detalhados e novas técnicas de deteção que consigam lidar com a variabilidade da fala. Este problema é a grande motivação desta dissertação, onde se pretende desenvolver técnicas de

deteção de eventos acústicos sobre gravações de crianças a ler de forma a alinhar o texto lido com o áudio gravado.

1.2 Objetivos

O objetivo desta dissertação consiste em identificar as palavras corretamente lidas e os eventos de disfluências, de forma a conseguir alinhar corretamente o áudio. Para isso, foi necessário desenvolver os modelos acústicos a partir de uma base de dados obtida através de gravações de crianças de várias escolas primárias. Após tentar alinhar o sinal de fala com o texto lido, assumidamente sem hesitações, o passo seguinte foi desenvolver técnicas para detetar eventos acústicos, adaptando a técnica de *wordspotting*, de forma a detetar eventuais erros de pronunciação ou disfluências no ato da leitura.

Este trabalho tem como base o projeto LestRead e visa melhorar os modelos acústicos para crianças, que foram obtidos através da adaptação de modelos de fala de adultos. Com isto pretende-se obter modelos mais robustos e que consigam obter resultados com um melhor desempenho.

1.3 Organização da Dissertação

A dissertação encontra-se organizada em 5 capítulos principais. Neste primeiro capítulo é feita uma introdução sobre o assunto desta dissertação, como a motivação a que nos levou à sua realização, objetivos e a sua estrutura. No capítulo 2, é feita uma contextualização sobre os modelos utilizados (HMM); sobre o funcionamento do HTK Toolkit, ferramenta utilizada no desenvolvimento desta tese; e sobre os métodos utilizados como redes neuronais e *wordspotting*. No capítulo 3 focamos o assunto do treino de modelos, onde começamos por explicar as bases dados utilizadas, os tipos de modelos desenvolvidos e alguns resultados obtidos. No capítulo 4 é feita a explicação do método sobre a deteção de palavras e identificação de erros de pronunciação através do *wordspotting*. Por último, no capítulo 5 são destacadas conclusões do trabalho realizado e enunciadas algumas sugestões para trabalho futuro.

Capítulo 2

Modelos Acústicos do Sinal de Fala

No reconhecimento de fala são utilizadas vários conceitos, ferramentas e técnicas, como é o caso de modelos HMM, HTK Toolkit, e *wordspotting*. Este capítulo fará uma breve introdução teórica aos modelos acústicos, HMM e ANN (redes neuronais artificiais), bem como aos métodos de descodificação do sinal acústico (com ferramentas do HTK Toolkit) e *wordspotting*, para ser possível entender o funcionamento destas ferramentas e técnicas e assim obter uma melhor compreensão do trabalho desenvolvido.

2.1 Modelos de Markov Não Observáveis

Os Modelos de Markov Não Observáveis (HMM) são processos duplamente estocásticos que se baseiam numa cadeia de Markov, mas em que cada estado da cadeia está associado uma função densidade de probabilidade. Dada uma sequência de observações gerada por um HMM, a sequência de estados implícita na sequência não é conhecida e, por isso, os modelos denominam-se como Modelos de Markov não observáveis [12].

Um HMM é constituído por um número de estados *N*. A cada instante *t*, um novo estado *j* é introduzido, gerando um símbolo de saída (observação), o_t. Cada estado j tem associado uma função de densidade de distribuição $b_j(o_t)$, que determina a probabilidade de gerar a observação o_t num instante de tempo *t*. Cada par de estados *i* e *j* tem associado uma probabilidade de transição a_{ij}. A *figura 2.1*, mostra um exemplo deste processo, onde um modelo de 6 estados percorre a sequência de estados X = 1,2,2,3,4,4,5,6 de formar a gerar a sequência o_1 a o_6.

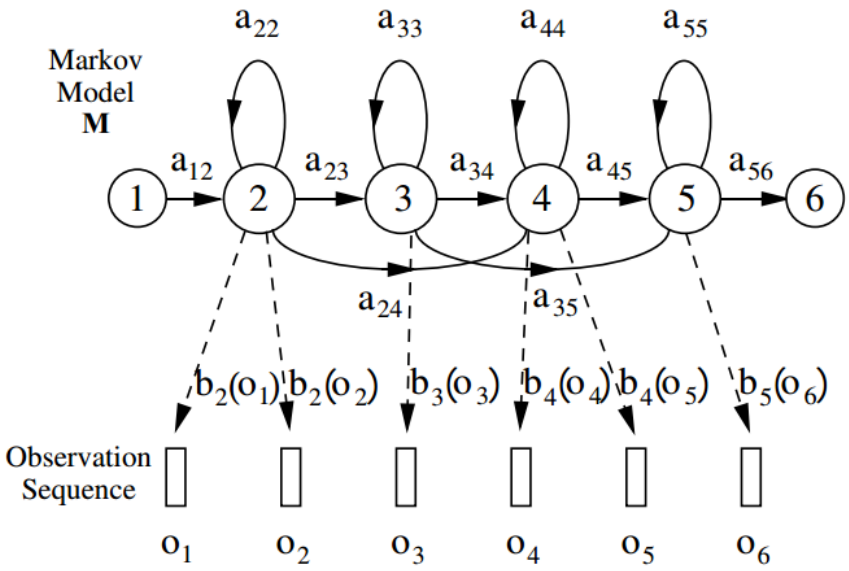

Figura 2.1 - Exemplo de um modelo de Markov. Retirado de [14]

A função densidade de distribuição $b_j(o_t)$ do símbolo de saída o_t de cada instante de tempo *t* e estado *j* é normalmente definida pela soma de M Gaussianas (mistura de Gaussianas ou simplesmente misturas)[12]:

$$b_j(o_t) = \sum_{m=1}^{M} \omega_{jm} \mathcal{N}(o; \mu_{jm}, \Sigma_{jm}) \tag{2.1}$$

onde *M* é o número componentes da mistura da distribuição, μ_{jm} e Σ_{jm} são a média e a matriz de covariância, respetivamente, das observações no estado j e ω_{jm} o peso próprio de cada uma das gaussianas na mistura, sendo a distribuição Gaussiana dada por:

$$\mathcal{N}(o; \mu_{im}, \Sigma_{im}) = \frac{1}{\sqrt{(2\pi)^L |\Sigma|}} \exp\left[-\frac{1}{2}(o - \mu_{im})^T \Sigma_{im}^{-1} (o - \mu_{im})\right] \tag{2.2}$$

onde *L* é o número de componentes do vetor de observações e *T* significa transposição matricial.

Nesta dissertação foram usados modelos de HMM esquerda-direita com apenas 3 estados, na cadeia de Markov, que são muitas vezes utilizados no reconhecimento de fala. Estes modelos apenas podem transitar para o estado seguinte ou permanecer no próprio estado (*figura 2.2*). Neste caso os modelos são para fonemas - as unidades elementares da fala, podendo dizer-se que os estados indicam o início, a parte estável e o fim do fonema. As observações são definidas em termos de vetores com informação espetral do sinal de fala.

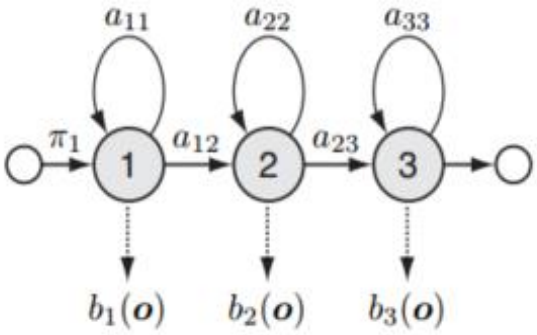

Figura 2.2 - Modelos esquerda-direita de 3 estados. Retirado de [13]

A modelação da fala com modelos de fonemas com contexto é muito mais robusta que a que não considera contexto. O contexto fonético consiste na identificação do(s) fonema(s) anterior(es) e do(s) seguinte(s) a uma dada realização de um fonema (fone), diferenciando os modelos consoante o contexto, ou seja, admitindo que existe coarticulação na realização dos fonemas. À modelação dos fones sem contexto chamamos monofones, enquanto que à diferenciação dos modelos com o contexto à esquerda e à direita chamamos de trifones.

O número de trifones é muito superior ao número de monofones. Usando um conjunto de 39 fonemas, poderemos ter milhares de trifones, em teoria poderiam ser 39^3=. Usualmente, na modelação com trifones, é feito um teste ao conjunto de trifones possíveis de forma a juntar modelos muito parecidos, usando técnicas de *clustering* (que serão abordadas mais à frente).

Como é observado na *figura 2.2*, os modelos têm 3 estados emissores e outros 2 que são ditos não emissores. Estes 2 estados não emissores apenas servem para concatenação entre HMMs, indicado o início e o fim de cada modelo. A matriz de transição destes modelos é uma matriz 5x5, em que a soma de cada linha é 1, exceto a última que é sempre zero uma vez que não são permitidas transições fora do estado final [14]. Para modelos transitórios opcionais, como pausas curtas e ruído, os modelos são considerados *tee models*, modelos que têm uma probabilidade transição diferente de zero do estado inicial para o final.

Na transformação do sinal de fala numa sequência de vetores utilizou-se a seguinte parametrização: logaritmos das energias à saída de um banco de 26 filtros com frequências centrais espalhadas numa escala mel (escala melódica que simula o funcionamento do sistema auditivo humana, onde a resolução em frequência diminui com a frequência, implicando que a largura de banda dos filtros aumenta com a frequência). A resposta em frequência dos filtros é triangular, desde a frequência de 150 Hz até 8000 Hz (metade da frequência de amostragem do sinal). As energias são calculadas multiplicando o quadrado do módulo da DFT (*Discrete Fourier*

Transform) de cada trama de sinal pela resposta triangular de cada filtro. As tramas são tomadas de 10 em 10 ms, segmentando o sinal com janela de Hamming de 25ms (400 amostras à frequência de amostragem de 16 kHz) às quais são aplicadas DFTs de 512 pontos. Depois de calculados os logaritmos das energias à saída de cada filtro do banco, segue-se uma transformação destes 26 parâmetros através da DCT (*Discrete Cossine Transform*) onde se pretende, de alguma forma, descorrelacionar os parâmetros iniciais. Além disso, esta transformação é propositadamente com perdas (são tomados apenas 12 parâmetros transformados) de forma a fazer alguma suavização espetral. Junta-se a estes 12 parâmetros o logaritmo da energia da trama, resultando no vetor, dito vetor MFCC (*Mel Frequency Cepstral Coefficients*) estático com 13 parâmetros, que representa uma trama do sinal acústico.

Finalmente, é muito usual juntar a estes 13 parâmetros os chamados coeficientes dinâmicos ou parâmetros delta, que consistem em tomar coeficientes de regressão linear no tempo de cada parâmetro estático, tomando uma janela de duas tramas antes e duas depois da trama considerada. São ainda adicionados os parâmetros delta dos parâmetros delta, chamados coeficientes de aceleração ou delta-delta. O vetor final tem assim 13x3=39 parâmetros, que são as observações a modelar com os HMMs.

2.2 HTK Toolkit

O HTK [14] é um conjunto de ferramentas de *software* para treinar e testar HMMs e é um sistema muito utilizado na área de reconhecimento de fala. Com este sistema é possível construir modelos HMM e fazer descodificação de fala - a conversão do sinal de fala numa sequência de eventos linguísticos (fonemas ou palavras, por exemplo) e que é o processo conhecido como reconhecimento automático de fala.

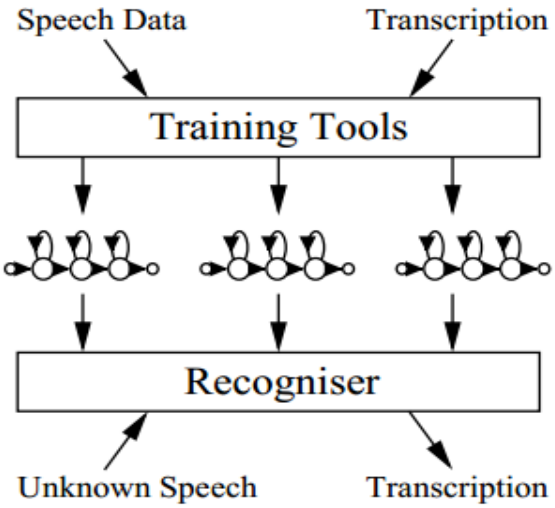

Figura 2.3 - Diagrama do funcionamento do HTK. Retirado de [14]

Como podemos observar na *figura 2.3*, o HTK é dividido em dois processos principais, o processo de treino e o processo de reconhecimento.

No processo de treino são estimados parâmetros de HMMs usando ficheiros de fala de treino e as suas respetivas transcrições, usualmente na forma ortográfica. O HTK fornece 5 ferramentas para estimar os parâmetros: *HCompV*, *HInit*, *HRest*, *HERest* e *HMMIRest*. *HCompV* e *HInit* são usadas para inicialização. A ferramenta *HCompV* define a média e a variância de cada Gaussiana de um HMM igual à média e variância global dos dados de treino. Uma alternativa mais detalhada, é iniciar com o *HInit*, calculando os parâmetros de um novo HMM usando o algoritmo de Viterbi que segmenta as observações por estados antes da estimação dos parâmetros do HMM. [14]. O *HRest* e o *HERest* são usados para refinar os parâmetros dos HMMs usando a reestimação *Baum-Welch* [14], algoritmo que encontra a estimativa da máxima verosimilhança dos parâmetros de um HMM.

Em geral, HMMs de palavras completas são criados através do *HInit* e do *HRest* enquanto HMMs de sub-palavras (fones) de fala contínua são criados através do *HERest*, inicializando com o *HCompV* ou *HInit* e *HCompV*, como mostra a *figura 2.4*. O *HMMIRest* é usado para treinar discriminativamente parâmetros de HMMs treinados.

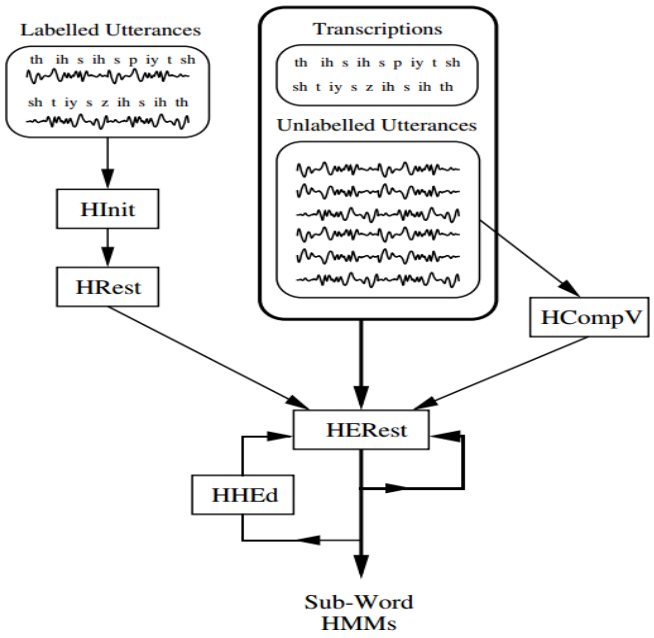

Figura 2.4 - Diagrama do treino de HMMs. Retirado de [14]

O processo de reconhecimento no HTK consiste na transcrição de ficheiros de fala. Este processo é controlado por uma rede de reconhecimento, que descreve a sequências de palavras que podem ser reconhecidas, um dicionário, que contêm as sequências de HMMs de cada palavra, e um conjunto de HMMs. O trabalho do descodificador é encontrar o caminho que tem maior probabilidade logarítmica. Para diminuir o tempo de processamento o descodificador apenas propaga *tokens* que têm alguma probabilidade de serem vencedores, a este processo chama-se *poda de caminhos*. Na descodificação é utilizada ferramenta *HVite* que permite fazer reconhecimento e alinhamento forçado. Para avaliar o desempenho do reconhecedor está disponível a ferramenta *HResutls* que faz um alinhamento ótimo entre a transcrição de referência e aquela fornecida pelo processo de reconhecimento e que, com este alinhamento, calcula as taxas de erro por símbolo do processo de reconhecimento

2.3 Rede Neuronal

Uma rede neuronal (Artificial Neural Network – ANN) é um grupo de nodos interconectados semelhante à rede de neurónios de um cérebro. Na *figura 2.*5, cada nodo circular representa um neurónio artificial e uma seta uma ligação a partir da saída de um neurónio para a entrada de outro. As forças das ligações entre neurónios são conhecidas como pesos sinápticos e estes são usados para armazenar o conhecimento experimental.

Estas redes são designadas de perceptrão multicamada (*Multilayer Perception* - MLP) [19], quando tem várias camadas e as ligações entre camadas são unidirecionais (*feedforward*, isto é, sem realimentação). A arquitetura destas redes consiste numa camada de entrada, constituída por um conjunto de nodos de entrada (nodos sensoriais), por uma ou mais camadas escondidas, constituídas por nodos de computação, e uma camada de saída, constituída por nodos que contêm a informação sobre a rede. A *figura 2.5* representa um exemplo da arquitetura descrita.

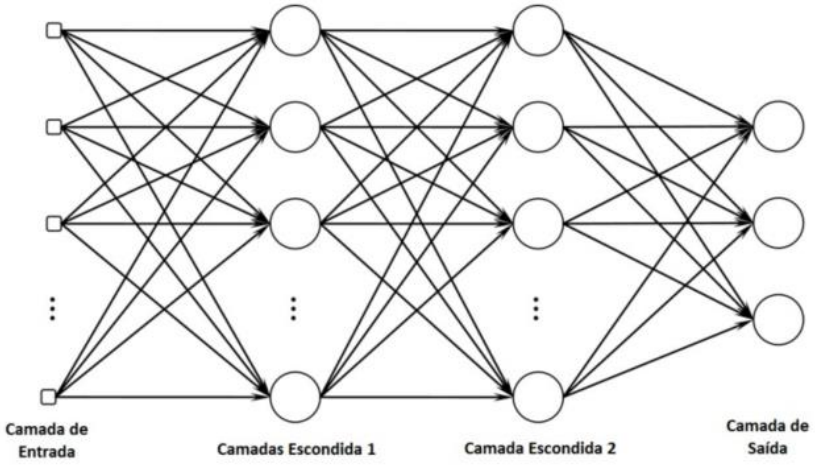

Figura 2.5 - Exemplo da arquitetura de uma MLP. Editado de [17]

O treino da rede baseia-se no algoritmo de retopropagação (*error back-propagation*) que consiste na propagação pelas diferentes camadas do erro verificado na camada de saída. O treino começa por propagar a reação à informação que é fornecida aos nodos sensoriais (perceptrões) das camadas de entrada, de camada em camada obtendo uma resposta na camada de saída que contêm o mapeamento da informação de entrada (transforma a informação acústica em informação linguística - fonemas, um por cada nodo de saída). A segunda etapa do algoritmo é propagar no sentido inverso a diferença entre a transcrição em termos de fonemas e o mapeamento obtido na resposta, corrigindo os pesos sinápticos de forma a aproximar a resposta da rede neuronal à resposta ideal.

Nesta dissertação, no *capítulo 4*, foi utilizado um sistema de reconhecimento de fonemas baseado no sistema proposto por [18]. Este sistema utiliza uma arquitetura entre HMMs, para uma descodificação de fonemas, e redes neuronais do tipo MLP.

O treino da rede tem como objetivo realizar o mapeamento de parâmetros de entrada para probabilidades *a posteriori* de fonemas em função da etiquetagem rigorosa dos mesmos. Este treino recebe vetores de coeficientes MFCC, em que o contexto temporal de cada um deles é dividida em duas partes, uma parte vai conter a informação do passado temporal e outra a

informação do futuro temporal. Os vetores são concatenados de acordo com o respetivo contexto e são fornecidos aos nodos de entrada de duas redes neuronais.

Estas duas redes neuronais têm a arquitetura MLP descrita anteriormente, com apenas uma camada escondida. Na camada de entrada as redes vão receber os vetores pré-processados, na camada escondida as redes têm as respostas à informação de entrada e na camada de saída é aplicada uma não linearidade de SoftMax [19], tornado a soma de todas as saídas unitárias, o que pode ser então interpretado como probabilidades *a posteriori* dos fonemas. Ao conjunto destes vetores de probabilidades em todas as tramas iremos chamar de posteriorgrama do sinal acústico. Os vetores de probabilidades são transformados para o seu logaritmo e novamente normalizados na média e na variância de forma a garantir que se encontrem todos na mesma gama dinâmica.

2.4 *Wordspotting*

Wordspotting é uma técnica que tem como objetivo procurar e encontrar uma *keyword*, (palavra ou parte de uma frase), num dado sinal de fala. Este método trata de um teste de hipóteses em que a hipótese nula, H_0 (hipótese considerada verdadeira), é "este sinal contém uma palavra-chave". Como erros de decisão, o sistema tem os *misses* ou falsas rejeições, quando o sistema rejeita a hipótese H_0 quando esta é verdadeira, e os *falses alarms* ou falsas aceitações, quando o sistema aceita a hipótese H_0 quando esta não é verdadeira. Esta hipótese é decidida mediante uma razão de verosimilhança - LR (Likelihood Ratio), caso LR seja superior ao limiar de aceitação o sistema aceita a hipótese H_0.

O modelo de *wordspotting* definido tem dois caminhos, *figura 2.6.* O *keyword model* calcula a verosimilhança da sequência de fonemas de uma palavra-chave num dado sinal. O *filler model* calcula a sequência máxima de verosimilhança dos fones dado o sinal e os modelos acústicas e serve para qualquer palavra-chave.

No caso em que a LR dos dois caminhos for aproximadamente igual (é sempre menor que 1) significa que a palavra-chave deve de estar contida no sinal de áudio. No caso de ser muito menor que 1, significa que a verosimilhança do caminho superior (*keyword model*) é muito baixa face à do caminho de baixo (*filler model*), logo a palavra-chave não está presente no sinal.

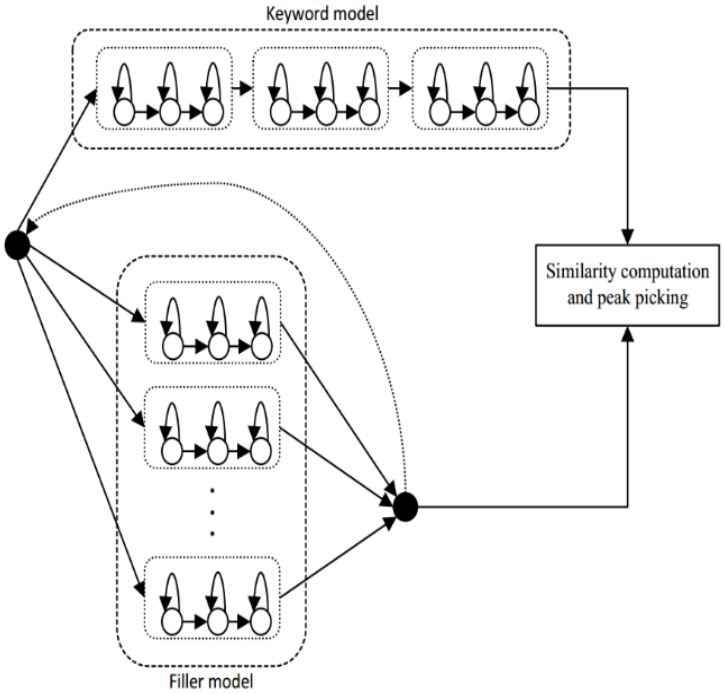

Figura 2.6 - Diagrama da descodificação através do Wordspotting. Retirado de [15]

Este sistema assume uma medida de semelhança baseada na razão de verosimilhança utilizando probabilidades logarítmicas (LLR – Log-Likelihood Rate) para evitar erros de precisão, dada por:

$$LLR(t) = \sum_{i=t-N(t)+1}^{t} LW(i) - \sum_{i=t-N(t)+1}^{t} LF(i) \tag{2.3}$$

onde $LW(i)$ e $LF(i)$ são as probabilidades logarítmicas do *keyword* e *filler model* na trama *i*, respetivamente.

Uma forma de obter uma razão de verosimilhança mais confiável é normalizar a razão anterior pelo número de tramas, *N(t).*

$$SS_1(t) = \frac{LLR(t)}{N(t)} \tag{2.4}$$

Outra normalização possível é baseada na duração média da *keyword*, $\overline{D}_W$.

$$SS_2(t) = \frac{LLR(t)}{\overline{D}_W} \tag{2.5}$$

Por último, uma normalização que aumenta a ocorrência de um pico num LLR obtém-se dividindo a razão de verosimilhança pela sua área [15].

$$SS_3(t) = \frac{LLR(t)}{A(t)} \tag{2.6}$$

$$A(t) = \sum_{i=1}^{N(t)} (LLR(t) - LLR(t-i)) \tag{2.7}$$

Capítulo 3

Criação de Modelos Acústicos de Fala de Crianças

A criação de modelos consiste na adaptação de modelos HMMs a uma base de dados, de forma a obter modelos mais robustos. A criação de modelos acústicos pode resultar através de um treino de raiz, ou da adaptação de modelos já existentes. Neste projeto foram desenvolvidos modelos para crianças baseados nos dois métodos. No caso do método de adaptação utilizou-se a adaptação MAP. No outro caso foram criados modelos de raiz, usando-se a inicialização de modelos com "flat-start".

3.1 Bases de Dados

Para desenvolver modelos acústicos é necessário uma base de dados de fala (ficheiros de áudio e respetivas transcrições). Nesta dissertação foram utilizadas duas bases de dados diferentes, CNG e LetsRead, ambas com gravações da fala de crianças.

3.1.1 Base de dados CNG

A base de dados CNG (*Contents for Next Generation Networks*) foi criada pela MLDC (*Microsoft Language Development Center*) e está descrita com algum detalhe em [16]. Esta base de dados contém cerca de 20 horas de fala portuguesa de crianças e está dividida em duas partes: CNG1 e CNG2. A parte do CNG1 contém as gravações da fala das crianças dos 3 aos 5 anos de idade enquanto que a parte do CNG2 contém dos 7 aos 10 anos.

Nesta dissertação apenas se utilizou a base de dados CNG2, isto porque as crianças com idade inferior a 6 anos têm uma maior dificuldade de leitura, o que prejudica o treino de modelos. Para desenvolver os modelos foram excluídas todas as locuções que não estavam em condições para treinar, por exemplo locuções com partes incompreensíveis e locuções com ruído em excesso. Com estas exclusões, esta base de dados contém cerca de 14 horas de fala, onde 6 horas

correspondem a sequências de números, 2 horas de números isolados, 1 hora de notas musicais e 5 horas de frases. Devido à pequena contribuição fonética nas notas musicais, estas também foram excluídas, obtendo assim uma base de dados final com cerca de 13 horas de fala. Do total 10% foi utilizado para teste e 90% para treino.

3.1.2 Base de dados LetsRead

A base de dados LestsRead contém gravações da leitura de crianças do 1º ao 4º ano de escolaridade. As gravações contêm frases e pseudopalavras [3]. Esta base de dados contém cerca de 6 horas e 30 minutos de fala e está dividida em duas partes, a base de dados da recolha de Julho de 2014 e a recolha de Dezembro de 2014. A primeira parte contém cerca 5 horas de fala enquanto que a de Dezembro contêm 1 hora e 30 minutos. Notar que apenas consideramos nesta base de dados os ficheiros etiquetados manualmente, o que corresponde a 104 crianças num total de 284.

As gravações da fala desta base de dados contém uma grande variedade de disfluências que representam os tipos de erros mais comuns na leitura das crianças do 1º ciclo do ensino básico. Para distinguir as várias disfluências na fala, foram criadas etiquetas distintas para cada tipo: PRE – caso em que a criança corrige o inico da palavra; SUB – caso em que uma palavra é substituída por outra; PHO – caso em que ocorre pronúncia errada, como a extensão de um fone ou a troca de um fonema; REP – caso em que a criança repete uma palavra; INS – existência de uma palavra extra que não se encontra na frase original; DEL – caso em que uma palavra não é pronunciada; CUT – caso de uma palavra é cortada, principalmente no início ou no fim, e que de seguida não é corrigida; EXT – caso em que ocorre uma extensão de um fonema; IWP – caos em que se regista uma pausa dentro de uma palavra, exemplo de uma palavra pronunciada sílaba por sílaba; NOI – caso em que ocorre ruído na gravação, ruído labial, de respiração ou mesmo de fundo.

Para desenvolver os modelos acústicos foi realizado o treino de apenas uma das partes da base de dados. Para treino foram selecionados 90% dos ficheiros de gravação da base de dados de Julho e apenas 10% para teste. A parte de Dezembro foi alinhada utilizando os melhores modelos obtidos resultante do treino da base de dados do CNG e da parte de Julho da base de dados do LetsRead.

3.2 Treino dos Modelos

Neste projeto começou-se por desenvolver modelos a partir do método *flat-start*, método que cria os modelos de raiz. Este método consiste na criação de modelos de monofones e a partir destes treinar os modelos de trifones, modelos com um maior contexto fonético.

3.2.1 Criação de HMMs de Monofones

A criação de HMMS de monofones começa por criar um modelo protótipo, que tem como propósito definir a topologia dos modelos. Os parâmetros de entrada deste modelo não são importantes. No caso de um sistema de fones uma boa topologia é a de 3 estados esquerda-direita, *figura 2.2*. A ferramenta *HCompV* irá receber este modelo protótipo e irá calcular a média e a variância global e criar modelos em que todas as suas Gaussianas tenham essa média e variância. Com os HMMs iniciais criados, os modelos são reestimados através da ferramenta *HERest*, que recebe como parâmetro um *pruning treshold*, que limita o número de estados que o algoritmo *forward-backward* inclui na soma e permite reduzir a quantidade de informação a ser computada, um ficheiro com as transcrições da fala em fones e a lista de fones utilizados. Este processo é ilustrado pela *figura 3.1*.

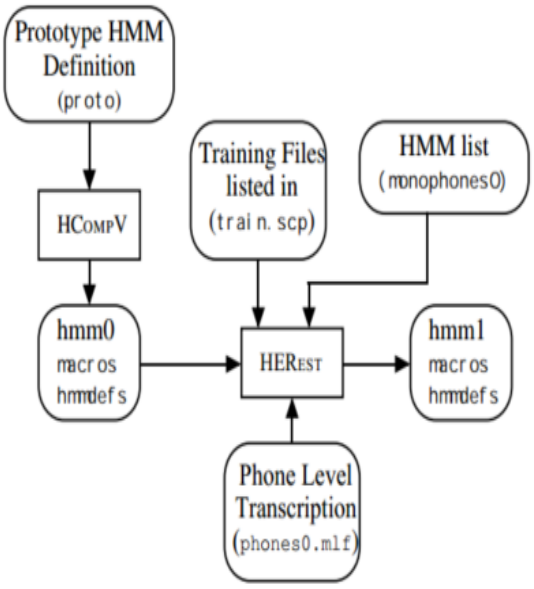

Figura 3.1 - Criação inicial dos HMMs de monofones. Retirado de [14]

O passo seguinte consiste em adicionar transições extra do estado 2 para o estado 4 e do estado 4 para o estado 2 no modelo de silêncio, *figura 3.2*, para o tornar mais robusto. Devido à existência de silêncios muito curtos entre palavras, criou-se o *tee-model* "sp" que representa os silêncios de duração curta ou nula. No final os modelos são novamente reestimados.

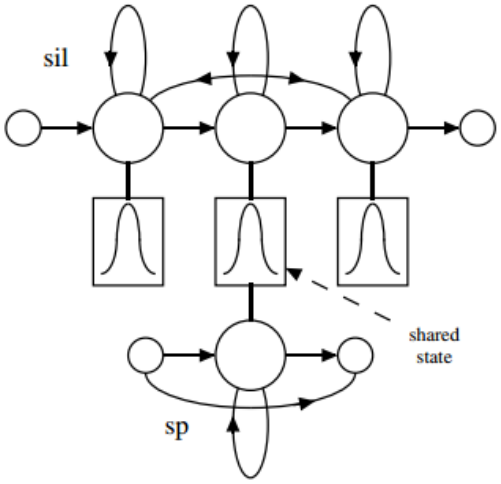

Figura 3.2 - Alteração do modelo de silêncio. Retirado de [14]

3.2.2 Criação dos Modelos de Trifones

Para criar modelos de trifones basta clonar os HMMs de monofones em modelos de trifones e re--estimar os modelos clonados usando transcrições em termos de trifones. Estas transcrições são criadas através da ferramenta *HLed* que cria uma lista com todos os trifones presentes nos dados de treino e cria as respetivas transcrições para cada ficheiro de áudio. Isto é, coloca contexto em cada fone previamente etiquetado.

Com a ferramenta *HHed*, que recebe um *script* que contém comandos para agrupar as matrizes de transição de cada fone, clonam-se os modelos. Agora os modelos de trifones têm matrizes de transição partilhadas. Como este método afeta a performance do treino é importante agrupar apenas parâmetros que tenha um pequeno efeito na discriminação. Estes são os casos em que os parâmetros de transição não variam significativamente com o contexto acústico, sendo necessário estimá-los com precisão.

Uma vez obtidos os modelos clonados e a transcrição em trifones, o novo modelo de trifones é reestimado usando novamente o *HERest*. No final desta reestimação, é gerado um ficheiro com as estatísticas de ocupação de estado. A *figura 3.3* ilustra o processo descrito.

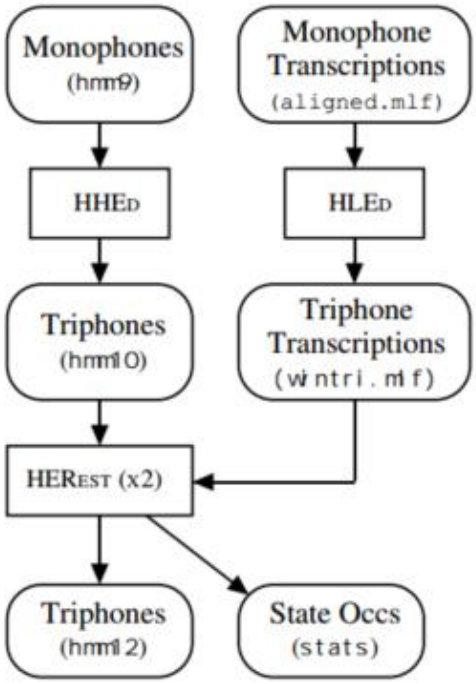

Figura 3.3 - Criação de modelos de trifones através de modelos de monofones. Retirado de [14]

O próximo passo consiste em agrupar os modelos de trifones muito parecidos, procurando formar *clusters* de estados de modelos HMM a partir de trifones com contexto semelhante. Para tal utilizou-se mais uma vez o *HHEd* mas desta vez recebe um ficheiro com comandos para fazer o *clustering*, usando árvores de decisão.

Uma árvore de decisão contém comandos que fazem questões sobre o contexto à esquerda e à direita de cada fone. A árvore aglomera todas as ocorrências de estados de modelos com esse contexto. Se a verosimilhança do conjunto aumentar esse grupo de estados é criado e é representado por um único estado que vai ser partilhado por todos os modelos em que está envolvido. As questões servem para criar hipóteses de agrupamento de estados. Apenas os grupos para os quais faz sentido a fusão é que são criados, caso contrário, a especialização da formação dos grupos continua (noutros nodos da árvore). Os *clusters* são criados à medida que as questões são respondidas e terminam quando se deixa de conseguir dividir o seu conteúdo. No final é criada uma *tiedlist*, que contêm a lista dos trifones agrupados e recorre-se novamente à reestimação *Baum-Welsh* dos modelos agrupados por *clusters* através do *HERest*. A *figura 3.4*, representa uma pequena parte de uma árvore de decisão. O *script*

QS “L_a” {a-*}

é um exemplo de uma pergunta que verifica se o estado atual representa um trifone com o fone [a] à esquerda, e é definida por “L_a”.

De notar que os modelos criados têm PDFs de apenas uma Gaussiana. O incremento do número de componentes para uma mistura Gaussiana é descrito na seguinte subsecção.

```
QS "L_V-Nasal" {ã-*,ë-*,ï-*,õ-*,ü-*,Ã-*,Ë-*,Ï-*,Õ-*,Ü-*}
QS "L_V-A-Nasal" {ã-*,ë-*,ï-*,õ-*,ü-*}
QS "L_V-T-Nasal" {Ã-*,Ë-*,Ï-*,Õ-*,Ü-*}
QS "R_V-Nasal" {*+ã,*+ë,*+ï,*+õ,*+ü,*+Ã,*+Ë,*+Ï,*+Õ,*+Ü}
QS "R_V-A-Nasal" {*+ã,*+ë,*+ï,*+õ,*+ü}
QS "R_V-T-Nasal" {*+Ã,*+Ë,*+Ï,*+Õ,*+Ü}

QS "L_a" {a-*}
QS "R_a" {*+a}
QS "L_á" {á-*}
QS "R_á" {*+á}
QS "L_a|á" {a-*,á-*}
QS "R_a|á" {*+a,*+á}
QS "C_á" {*-á+*,á+*,*-á,á}
```

Figura 3.4 – Exemplo de uma árvore de decisão

3.2.3 HMMs de Mistura de Gaussianas

Finalmente, converteram-se os HMMs de uma Gaussiana para HMMs com PDFs definidas com uma mistura de Gaussianas, utilizando o comando *HHEd*. Este comando utiliza um processo denominando de *mixture splitting* [14], que incrementa o número de misturas, dividindo repetitivamente a componente ‘mais pesada’ até que se obtenha o número de misturas pretendido. Os comandos utilizados para este processo foram

MU *n* {*.state[2-4].mix}

que incrementam o número de componentes em todas as misturas ao mesmo tempo até *n*, do estado 2 ao estado 4. Por cada mistura realizada, os modelos óbitos foram reestimados com o comando *HERest*.

Deste processo de treino resultaram os modelos acústicos da base de dados do CNG e do LetsRead.

3.3 Adaptação de Modelos

Após a criação de modelos para a base de dados CNG, considerou-se interessante usar um processo de adaptação dos modelos criados às condições acústicas da base de dados Letsread.

Optou-se por adaptar os melhores modelos obtidos à parte de treino da base dados do LetsRead, para que fosse possível comparar com os modelos criados para o LetsRead. Para tal utilizou-se o método de adaptação MAP, *maximum á posteriori* [14].

Este método de adaptação necessita do conhecimento prévio sobre a distribuição dos parâmetros dos modelos. A nível matemático este método resume-se a uma simples forma de adaptação do estado *j* da mistura *m*:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau}\bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau}\mu_{jm} \tag{3.1}$$

onde τ é o coeficiente de ponderação do conhecimento *á priori* dos dados de adaptação, *N* é a verosimilhança de ocupação dos dados de adaptação, definida por (*3.2*), μ_{jm} é a média dos dados iniciais e $\bar{\mu}_{jm}$ é a média dos dados de adaptação observada definida por (*3.3*).

$$N_{jm} = \sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t) \tag{3.2}$$

$$\bar{\mu}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t) o_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)} \tag{3.3}$$

Como se pode verificar nas fórmulas anteriores, se a verosimilhança de ocupação da componente Gaussiana (N_{jm}) for baixa, então a estimativa média MAP permanecerá perto da média independente. Outros casos a verificar, são os casos de $\tau = 0$ e $\tau = \infty$, que representam o caso em que os parâmetros ficam iguais aos dos dados iniciais e caso em que ficam iguais aos dos dados adaptados, respetivamente. Com a adaptação MAP, cada componente é atualizada com uma estimativa MAP, com base na média á *priori*, na ponderação e adaptação de dados.

Para executar esta adaptação, recorreu-se ao comando *HERest* que fornece a opção que permite adaptar a média e a variância para uma dada adaptação. Neste processo foram utilizados os modelos de 9 e 16 misturas da base de dados do CNG.

3.4 Resultados

Com o treino de modelos concluído, realizaram-se vários testes nas várias bases de dados com o objetivo escolher os melhores modelos para este projeto.

Para realizar os testes, utilizaram-se as ferramentas de descodificação fornecidas pelo HTK, *HVite* e *HResults*.

O descodificador *HVite* recebe uma *task-garmmar*, isto é, a definição de todas as sequências possíveis de serem reconhecidas. Nesta dissertação foram utilizadas dois tipos de *task-garmmar*, uma gramática livre, em que qualquer sequência é possível, todos os fones têm peso 1, e uma bigrama, que tem história, isto é define a probabilidade de ocorrência de dois símbolos seguidos, *i* e *j*. Aqui os pesos já não podem ser 1.

Para conseguir uma melhor avaliação dos modelos obtidos, todos os testes foram realizados com vários valores de penalização de inserções, para evitar que os modelos tenham um número de inserções muito superior que apagamentos e vice-versa.

Os resultados apresentados representam a percentagem de palavras corretas (3.4) e a sua exatidão (3.5):

$$\%Correct = \frac{H}{N} \times 100\% \tag{3.4}$$

onde *H* corresponde ao número de palavras/fones corretos e *N* o número total de palavras.

$$\%Accuracy = \frac{H - I}{N} \times 100\% \tag{3.5}$$

onde *I* são o número de inserções existentes nos dados descodificados.

3.4.1 Teste CNG

A primeira base de dados a testar foi a CNG. Para tal foi necessário executar testes empíricos de penalização de inserção com ficheiros de teste da própria base de dados, para escolher o melhor modelo.

Como esta base de dados contém ficheiros de fala com conteúdo linguístico muito diferente decidiu-se realizar vários testes com diferentes partes da base de dados. Realizaram-se testes com gramática livre para ficheiros com apenas sequências de números e ficheiros de números isolados. No caso de ficheiros apenas com frases utilizou-se uma bigrama e uma gramática constituída apenas com as frases da base de dados e assim só possibilitava a construção dessas frases. Esta última gramática consistia numa rede que colocava todas as frases em paralelo, e que no processo de descodificação apenas uma delas era escolhida. Para a base de dados completa utilizaram-se os dois tipos de rede: gramática livre e bigrama.

Tabela 3.1- Resultados da descodificação do CNG

Base de Dados	*Valor de penalização*	*Nº de misturas*	*%Correct*	*%Accuracy*
Sequências de números	-20	16	98.17	98.01
Números isolados	-10	16	98.32	98.17
Frases (gramática de frases)	-5	13	99.27	98.70
Frases (bigrama)	-20	16	91.10	86.72
Base de dados completa (gramatica livres)	-20	10	75.38	63.40
Base de dados completa (bigrama)	-20	9	80.04	73.23

Com os resultados dos modelos das várias misturas escolheram-se os melhores desta base de dados para serem adaptados à base de dados do LetsRead. Os modelos escolhidos foram os de 9 e 16 misturas.

Como se pode observar na *tabela 3.1*, os modelos de 16 misturas foram os que tiveram melhores resultados mais frequentemente, justificando assim a sua escolha. No teste de gramática livre para a base de dados completa os modelos vencedores foram os modelos de 10 mistura, mas com uma diferença muito pequena dos modelos de 9 misturas. Para escolher o melhor dos dois testou-se com a mesma base de dados mas com uma rede bigrama e os modelos vencedores foram os de 9 misturas.

3.4.2 Teste LetsRead

Com os modelos do CNG adaptados e os modelos treinados com a base dados LetsRead, realizaram-se testes de descodificação de fones e de palavras no LetsRead. A gramática utilizada foi uma gramática livre, isto é, qualquer fone ou palavra pode suceder a qualquer um outro.

As tabelas seguintes mostram os resultados obtidos.

Tabela 3.2 - Resultados da descodificação de fones do LetsRead

Base de Dados	*Valor de penalização*	*Nº de misturas*	*%Correct*	*%Accuracy*
CNG	-100	1	57.830	42.370
CNG MAP	-100	2	68.410	57.220
LetsRead	-40	13	75.150	64.420

Tabela 3.3 - Resultados da descodificação de palavras do LetsRead

Base de Dados	*Valor de penalização*	*Nº de misturas*	*%Correct*	*%Accuracy*
CNG	-100	1	22.650	9.750
CNG MAP	-70	1	34.370	23.140
LetsRead	-30	9	33.700	25.270

Como era de esperar os modelos do CNG que não foram adaptados obtiveram os piores resultados. Desta forma, foram escolhidos os dois modelos adaptados e os dois melhores modelos do LetsRead, o melhor para o caso de fones (13 misturas) e o melhor para o caso de palavras (9 misturas), para serem aplicados no sistema de deteção de repetições de palavras (REP) e falsos inícios de palavras (PRE) no âmbito do projeto Letsread, [3], e que é abordado de seguida.

3.4.3 Deteção Automática de Disfluências Utilizando Gramáticas Específicas

Sendo as repetições (REP) e os inícios corrigidos (PRE) as duas disfluências mais comuns na fala das crianças na base de dados LetsRead (mais de 50%), este método apenas foca na sua deteção. Isto significa que todos os outros erros substituídos pela palavra correta e todos os silêncios no meio de palavras foram retirados.

Um erro comum, especialmente em crianças do 1º ano do ensino básico, é pronunciar uma palavra sílaba a sílaba, o que leva a pausas entre sílabas. Estas pausas são um grande problema para os descodificadores automáticos que não contam com elas. Para contornar este problema, este sistema encontra e corta os segmentos de sinal de baixa-energia criando novos sinais, e a descodificação é realizada com estes novos sinais.

Nesta descodificação cada gravação de fala tem uma gramática específica, de forma a detetar disfluências. Esta gramática consiste numa rede que tem a sequência original das palavras. No entanto, entre cada palavra existe uma sub-rede que permite a ocorrência de REPs e de PREs entre cada palavra, *figura 3.5 a)*. Nestas sub-redes, também são permitidas pausas entre a ocorrência de disfluências e as palavras, que representam silêncios, respiração e ruído. Além disso, as sequências de palavras são por vezes repetidas ou corrigidas. Para que este evento seja possível, são introduzidas sub-redes na rede principal permitem voltar atrás uma ou duas palavras, *figura 3.5 b)*.

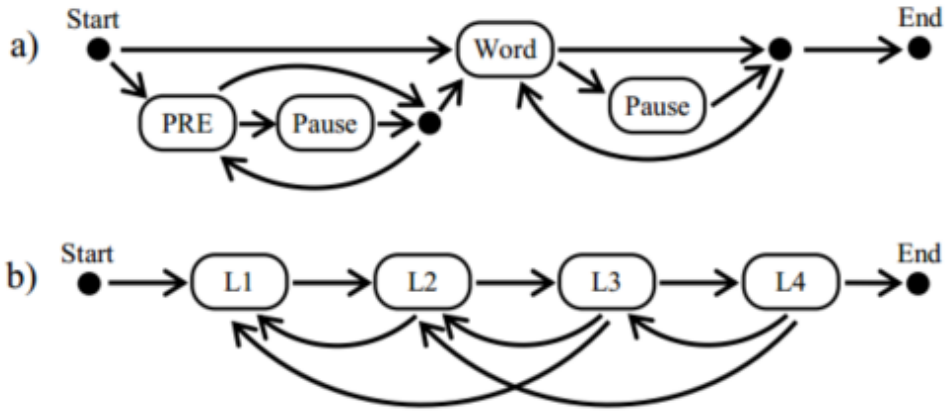

Figura 3.5 - Esquema de uma sub-rede de uma palavra (a) e rede de 4 palavras utilizando sub-redes (b). Retirado de [3]

Para fins de comparação, o sistema base é considerado o sistema de alinhamento forçado da sequência original das palavras, onde não são consideradas as disfluências, e para o caso do sistema

final é considerada uma probabilidade de inserção que conduz a cerca de 2.6% de taxa de falsos alarmes. O primeiro teste, com resultados na *tabela 3.4*, compara a taxa de acerto dos dois sistemas independente do contexto temporal, enquanto o segundo, *tabela 3.5*, compara a taxa de acerto mas com contexto temporal, variando o limite de variação - colar temporal.

Tabela 3.4 - Percentagem de palavras corretas e exatidão

	Modelo Básico	*Sistema Final*
%Correct	92.18	97.81
%Accuracy	91.86	94.80

Tabela 3.5 – Percentagem de segmentos com correspondência de etiqueta e limites com diferentes colares de tolerância

	Modelo Básico	*Sistema Final*
Colar de 50ms	55.72	58.43
Colar de 100ms	70.05	82.65
Colar de 150ms	76.05	89.39
Colar de 200ms	79.99	92.29
Colar de 250ms	82.76	93.69

Analisando a *tabela 3.4*, verifica-se que o método proposto supera o sistema base, isto porque este, por definição, não deteta os eventos REP nem PRE. No caso da *tabela 3.5*, podemos verificar que apenas com um colar acima de 100ms o sistema consegue ter uma taxa de acerto aceitável. Existem alguns fatores que podem explicar esta discrepância do alinhamento com as referências manuais. As anotações manuais podem diferir de 100ms e por vezes as anotações de pausas foram feiras com demasiado detalho (por vezes com menos de 3 tramas) o que origina muitos apagamentos de etiquetas e influenciam o resultado do sistema antes e depois de uma pausa, mesmo que curta.

Com o objetivo de avaliar a performance do sistema final na deteção de PREs e REPs, foram considerados como falsa aceitação (*false alarme*) qualquer deteção que não está alinhada com o alinhamento de referência, e como falsa rejeição (*miss*) qualquer REP ou PRE que não foi detetado e também qualquer substituição ou deteção errada. Usando diferentes penalizações de inserções, obteve-se uma variação nas falsas rejeições e nas falas aceitações, representados pela curva de *Detection Error Tradeoff* (DET) da *figura 3.6*.

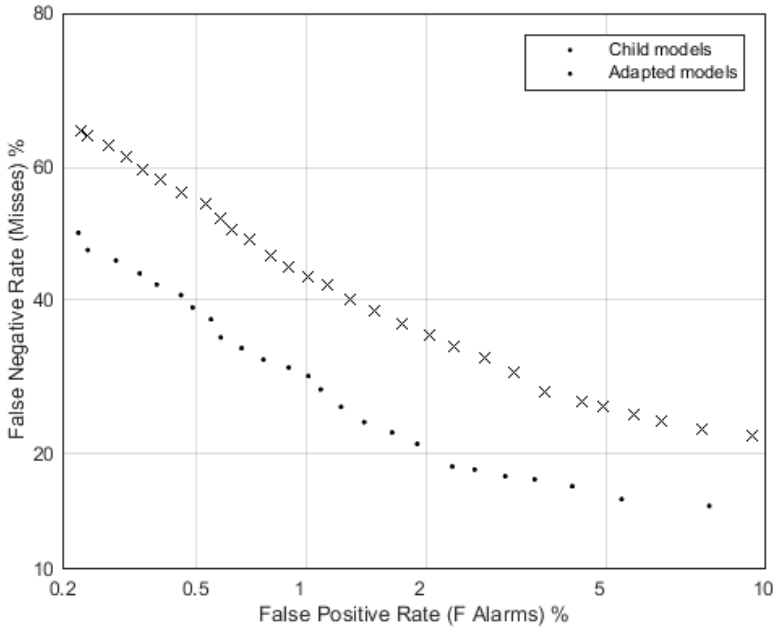

Figura.3.6 - DET do sistema de deteção de REPs e PREs variando a probabilidade de inserção de uma palavra

Comparando a curva dos modelos do LetsRead com a curva dos modelos adaptados, verifica-se que, para aproximadamente 5% de falsas aceitações, se obtém cerca de 15% de falsas rejeições no LetsRead e 25% nos modelos adaptados, o que mostra que os modelos treinados diretamente com a base de dados de trabalho são muito mais robustos.

Capítulo 4

Deteção de Palavras e de Erros de Pronunciação

Para desenvolver um sistema de avaliação da capacidade de leitura de crianças é necessário dispor de um sistema de reconhecimento de fala capaz de detetar disfluências, nomeadamente palavras como repetições e erros de pronunciação. Erros de pronunciação podem consistir, por exemplo, em sílabas não ditas ou trocadas. Neste âmbito criou-se um sistema baseado no conceito de *Wordspotting* descrito no *capítulo 2.4*. Assim, através deste método é possível verificar se uma certa sequência de palavras foi dita corretamente ou se existe algum erro, como erros de pronúncia ou repetições.

No desenvolvimento deste sistema, utilizou-se a base de dados do LetsRead completa, com os ficheiros de treino e de teste, onde o modelo acústico consiste na Rede Neuronal.

4.1 HMMs vs. Rede Neuronal

Com o decorrer em paralelo no laboratório de Processamento de Sinal da dissertação [18], aproveitou-se o método desenvolvido pelo autor de treino de uma rede neuronal para criar modelos de fones em português europeu de fala de crianças. A grande vantagem de se usar uma rede neuronal é que as saídas podem ser consideradas como probabilidades *á posteriori* dos fonemas dado o sinal acústico de entrada. Por outro lado, os HMMs apenas modelam as PDFs do sinal acústico, sabendo que estamos na presença de um dado fone, e é difícil calcular a partir destas PDFs as probabilidades *á posteriori* dos fonemas. Além disso, a robustez da rede neuronal é muito superior à dos HMMs quando se calculam as probabilidades *á posteriori*. Tal foi verificado no teste que se descreve de seguida.

Os modelos HMMs descritos anteriormente são modelos de trifones. Para ser possível calcular as probabilidades *á posteriori* dos fones foi necessário passar de trifones para fones sem perder o contexto fonético. Para tal, criou-se um dicionário de fones com multipronunciação - isto é, onde cada fone pode ser pronunciado de tantas maneiras quantos os trifones desse fone existam (todos os trifones com o mesmo fone central do trifone). Obtemos assim uma descodificação de fones através de modelos de trifones. De seguida faz-se uma descodificação em termos de fones

usando os modelos HMM e o modelo híbrido HMM/ANN. A *tabela 4.1* mostra os resultados obtidos em cada modelo.

Tabela 4.1 - Resultados dos modelos HMMs e dos modelos da Rede Neuronal

	HMMs	*Rede Neuronal*
%Correct	53.52	76.27
%Accuracy	45.00	72.25

Após a análise dos dados anteriores verificou-se que os modelos criados pela rede obtêm melhores resultados. Como o sistema desenvolvido tem como objetivo verificar a existência de uma determinada sequência de fones, os modelos utilizados têm de ser os que têm uma maior taxa de acerto, que neste caso são os novos modelos criados pela rede.

Seguidamente decidiu-se efetuar algumas alterações nos modelos para ficarem mais próximos da realidade. A primeira alteração consistiu na alteração do modelo "@" (fonema em *temer* ou *feliz*). Este modelo obrigava que a descodificação gastasse no mínimo 3 tramas, mas na fala contínua, o fonema representado por este modelo muitas vezes é 'apagado' (elidido), isto é ao falar este fonema pode quase não ser pronunciado. Para tornar este acontecimento possível o modelo foi alterado permitindo que o modelo consumisse apenas uma trama (no estado 1, ver *figura 4.1*) em alternativa a gastar 3 ou mais tramas.

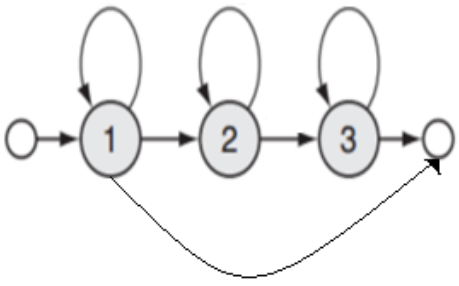

Figura 4.1 - Representação do modelo '@' alterado

A segunda alteração foi a inclusão de um modelo para pausas pequenas, o modelo "sp" (short pause). Para tal foi criado um modelo com os estados iguais aos do modelo "sil". A diferença entre este modelo e o modelo "sil" reside na matriz de probabilidades de transição, tal como indicado na *figura 4.2* É opcional passar por 1, 2 ou 3 estados, dependendo do tamanho do silêncio, ou então não passar por nenhum estado (não consumir nenhuma observação), quando não existir pausa (*tee-model*).

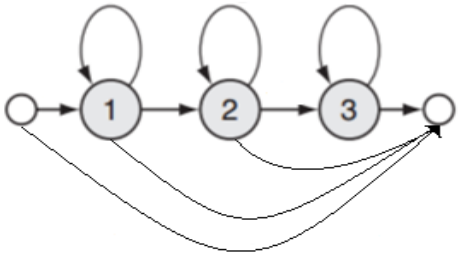

Figura 4.2 - Representação do modelo 'sp'

4.2 Medida de Semelhança

Antes de começar com a deteção de erros foi necessário escolher a medida de semelhança adequada ao sistema utilizado, tal como foram definidas no *capítulo 2.4* (escolha entre o LLR e uma das suas normalizações).

Uma das adaptações necessárias ao sistema foi o cálculo da duração média dos fones. No sistema original do *wordspotting*, a duração dos fones era obtida pelas matrizes de transição, mas aqui, nos modelos da rede, as matrizes não são estimadas. Para calcular essa duração foi criado um algoritmo que procura todas as ocorrências de cada fone na nossa base de dados, obtendo assim uma melhor exatidão no cálculo da segunda normalização do LLR, *SS2*. A *tabela 4.1* mostra a duração média de cada fone.

Tabela 4.2 - Duração média de cada fone

Fones	*Duração média (ms)*	*Fones*	*Duração média (ms)*	*Fones*	*Duração média (ms)*
@	114	*J*	153	*r*	97
&	145	*k*	94	*R*	125
a	165	*l*	120	*s*	176
b	90	*L*	130	*S*	152
d	95	*m*	107	*sil*	469
e	146	*n*	118	*t*	105
E	173	*&N*	146	*u*	136
eN	186	*o*	152	*uN*	173
f	175	*O*	150	*v*	113
g	95	*oN*	182	*z*	128
i	145	*p*	80	*Z*	145

Para avaliar o comportamento do *wordspotting* decidiu-se fazer 5 testes diferentes, variando o número de fones da palavra-chave a pesquisar, para fazer uma escolha da medida de semelhança mais ao pormenor. Começou-se por testar 10 palavras-chave contendo entre 10 a 15 fones. Como a nossa base de dados não tinha muitas frases repetidas, foram escolhidas sequências que apareciam pelo menos 3 vezes, dando assim um total de 31 ocorrências para um total de 22680 ficheiros de fala. O gráfico seguinte mostra as DETs em função da taxa de falsas aceitações (*False Accept Rate* – FAR) e taxa de falsas rejeições (*False Reject Rate* – FFR) das 4 medidas de semelhança.

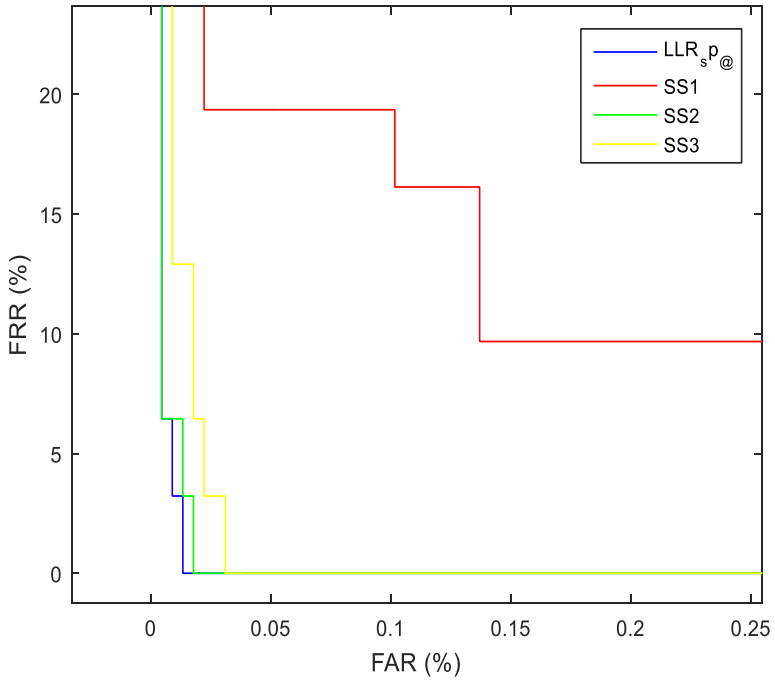

Figura 4.3 - DET de sequências de 10 a 15 fones

Analisando a *figura 4.3*, podemos concluir que a melhor medida de semelhança nesta situação é o LLR com uma taxa de 0.013% de falsas aceitações e 0% de falsas rejeições. Como seria de esperar a taxa de falsas rejeições é 0% devido ao facto de terem sido escolhidas sequências com um grande número de fones, considerando o tamanho de cada frase, tornando assim estas sequências praticamente únicas pois são muito diferentes do resto do contexto fonético das frases.

Para o teste seguinte foram escolhidas 10 palavras-chave contendo entre 6 a 9 fones. Neste caso, como a probabilidade de repetição é maior, foram escolhidas sequências com entre 10 a 30 ocorrências, o que resultou num total de 163 ocorrências para um total de 22680 frases.

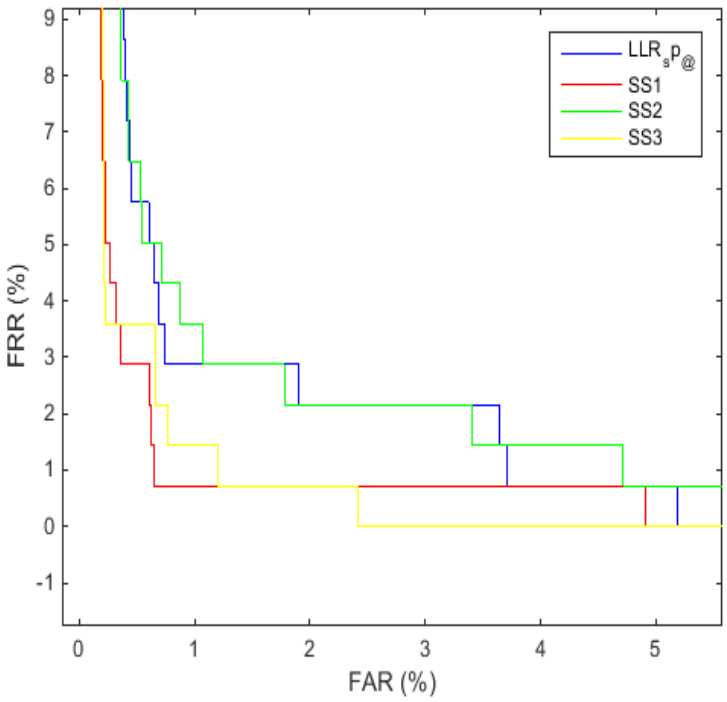

Figura 4.4 - DET de sequências de 6 a 9 fones

Neste caso, a normalização do LLR pelo número de tramas, *SS1*, obtém o melhor resultado, 6.8% de falsas aceitações e 7.1% de falsas rejeições. É normal esta DET obter resultados de semelhança com taxas superiores ao anterior, pois estas sequências são mais curtas o que provoca um aumento nas sequências semelhantes, tornando a tarefa de descodificação mais complicada para o sistema.

Também foram testados os casos de palavras-chave com um número elevado de fones, 16 a 20, e o caso de palavras-chave com um número muito reduzido de fones - de 3 a 5. Para o primeiro caso existiam 9 ocorrências num total de 6804, devido à falta de repetição das frases, enquanto no segundo caso existiam 855 ocorrências num total de 22680 frases.

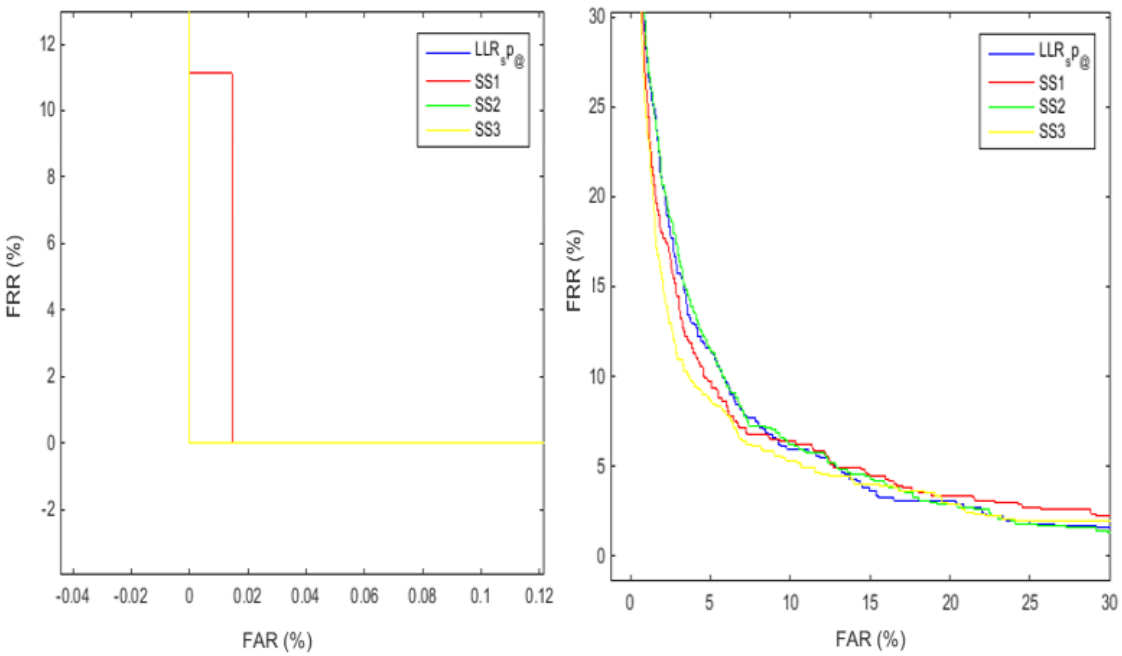

Figura 4.5 - DET de sequências de 16 a 20 fones (esquerda) e DET de sequências de 3 a 5 fones (direita)

Nestes dois casos ocorrem dois resultados totalmente diferentes. O primeiro para o caso de sequências de fones entre 16 a 20, tem um acerto de 100% exceto na medida *SS1*. Esta taxa de acerto é possível pois as frases da nossa base de dados são distintas, pelo que, com este número de fones, não existe grande probabilidade de confusão entre sequências por parte do sistema. Já no caso seguinte, acontece o inverso, as sequências são tão pequenas que o sistema tem grandes dificuldades de descodificação, obtendo assim um ponto próximo do de igual erro com uma taxa de 6.7% de falsas aceitações e 6.6% de falsas rejeições.

Por último, devido ao facto dos testes anteriores reunirem poucas ocorrências e o de 3 a 5 fones resultar num número elevado de erros, resolveu-se fazer um teste com todas as sequências anteriores para decidir uma medida de semelhança geral. Resultando em 1058 ocorrências num total de 74844 frases.

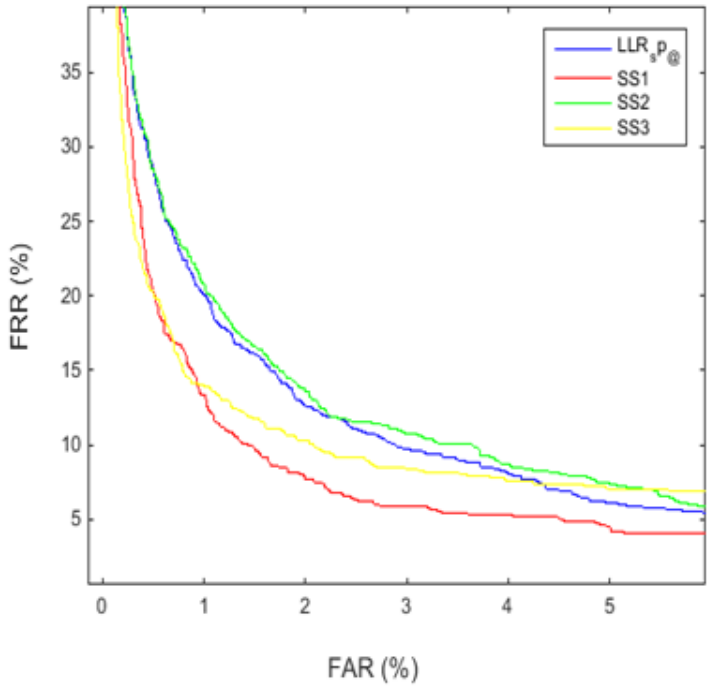

Figura 4.6 - DET de todas as sequências usadas

Para este caso a melhor medida de semelhança é a normalização do LLR pelo número de tramas, *SS1*, que obtém assim um ponto próximo à mínima distância à origem com uma taxa de 3.35% de falsas aceitações e 5.37% de falas rejeições.

Exceto nos estudos de sequências de 3 a 5 fones e com todas as sequências, os resultados obtidos não são considerados resultados confiáveis pois contêm muito poucas ocorrências, resultando em taxas de erro pouco fiáveis. A medida *SS1*, a normalização do LLR pelo número de tramas, foi assim escolhida como medida de referência para o desenvolvimento do sistema seguinte, pois obtém um número superior de ocorrências e uma taxa de acerto aceitável para o nosso estudo.

4.3 Sistema de Deteção de Disfluências

Este método começa por procurar uma palavra-chave num determinado ficheiro de fala devolvendo o LLR normalizado, *SS1*, escolhido anteriormente. Com essa medida de semelhança o sistema procura todos os picos superiores a um determinado limiar e separados entre eles por uma distância superior à metade da duração média da sequência de fones a procurar. Isto para garantir que todos os picos escolhidos estão relacionados com momentos de fala diferentes.

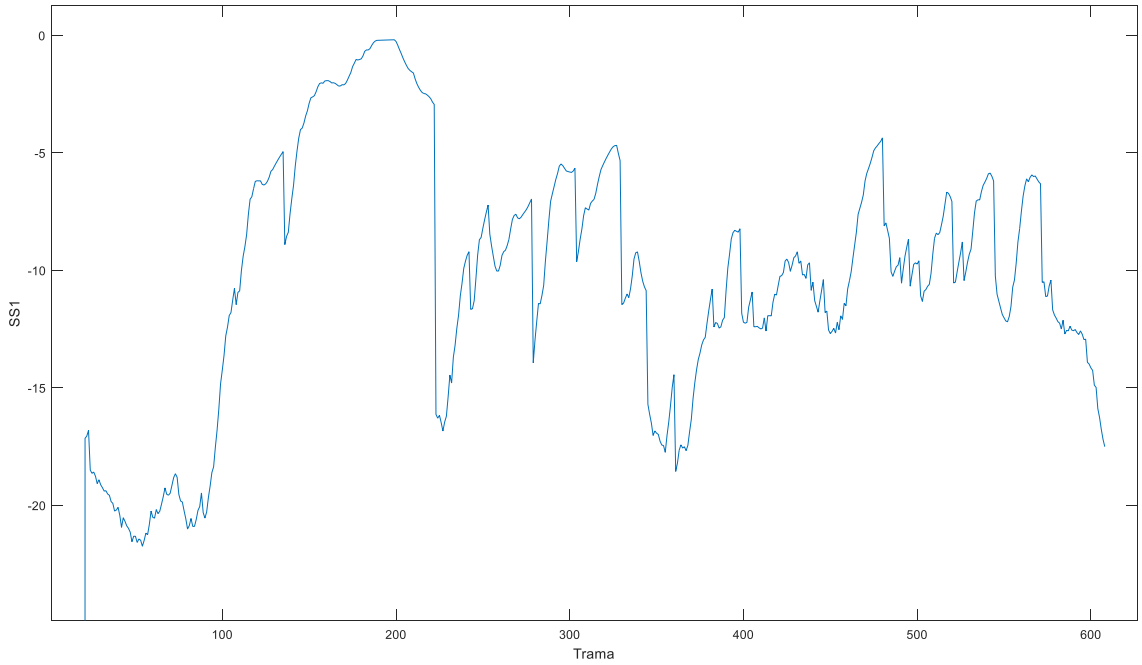

Figura 4.7 - Curva do LLR normalizado pela divisão do número de tramas, SS1, para a palavra-chave "coelhinho"

Como se pode verificar na *figura 4.7*, na zona da trama 200 ocorre o maior pico e o único acima do limar, -1.5 (limiar usado para o estudo deste sistema). Então podemos decidir que este sinal de fala contém a palavra "coelhinho", mas sem garantias de que foi dita corretamente.

Para verificar se não existem erros de pronunciação recorreu-se aos posteriorgramas fornecidos pela rede neuronal. Com o posteriorgrama da locução em causa e a sequência de fonemas correspondente ao texto lido, é necessário verificar se as duas sequências de fones conferem. O sistema verifica se numa determinada trama, a probabilidade do fone esperado é superior à de todos os outros, caso se verifique assumimos que esse fone se encontra naquela trama. Para o caso contrário, o sistema determina a diferença entre a probabilidade do fone com probabilidade superior e probabilidade do fone esperado. Se a diferença for superior a 0.3 assumimos que o fone não foi dito corretamente, se for inferior assumimos que existe uma dúvida devido à pronúncia do fone desejado ser muito semelhante a outro fone.

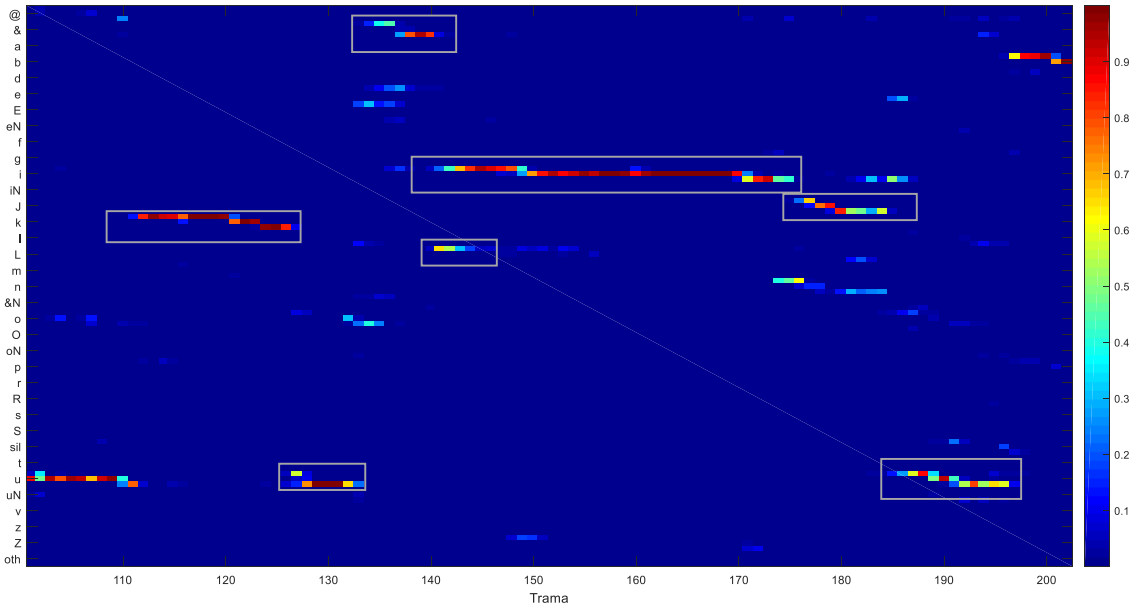

Figura 4.8 - Posteriorgrama que contém a palavra "coelhinho"

Na *figura 4.8*, podemos verificar que os fones da palavra "coelhinho", [*k u & L i J u*], estão com cores mais fortes, o que indica que a probabilidade desses fones é a probabilidade superior nessas tramas. Assim podemos concluir que o ficheiro de fala contém a palavra e que essa palavra foi dita corretamente.

4.4 Resultados

Com o método pronto a testar, foram realizados vários testes para várias situações com o objetivo de verificar a eficiência deste método e destes modelos. Para a realização destes testes foi escolhido um limiar mínimo de -1.5 para a escolha dos picos possíveis do LLR.

Começou-se por realizar 2 testes para a situação em que não ocorre nenhum erro para assim verificar a exatidão do sistema. No teste 1 procurou-se pela palavra "atrapalhada" no ficheiro que continha a frase seguinte: "Uma história muito atrapalhada". Na *figura 4.9* podemos verificar que o sistema encontra um pico que se destaca dos outros, e que se encontra próximo do intervalo de tempo da palavra verificado manualmente. Na análise do posteriorgrama, *figura 4.10*, o sistema indica que a palavra se encontra no ficheiro e não tem nenhum erro de pronunciação.

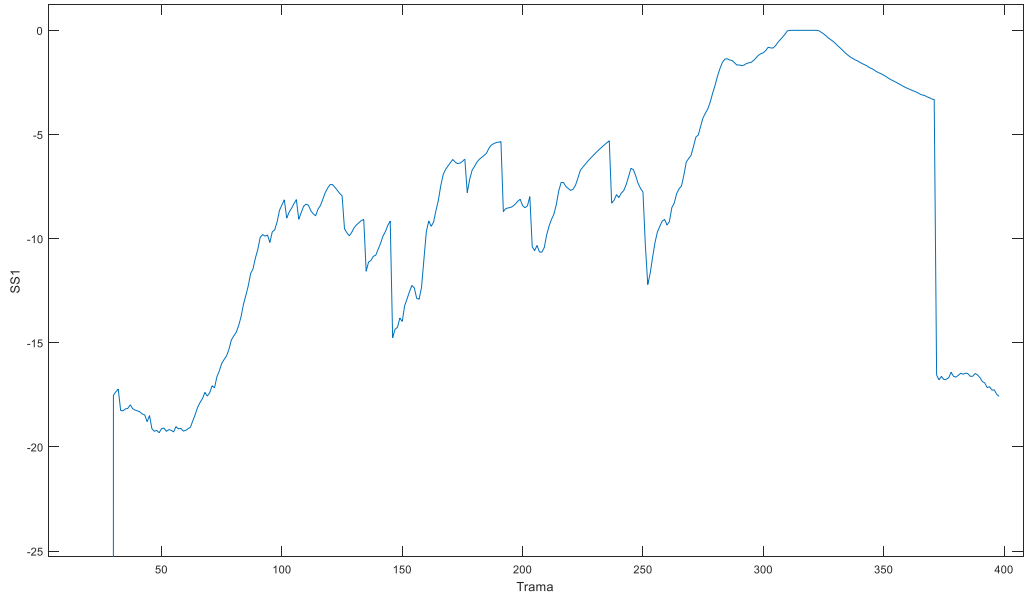

Figura 4.9 - LLR normalizado do teste 1

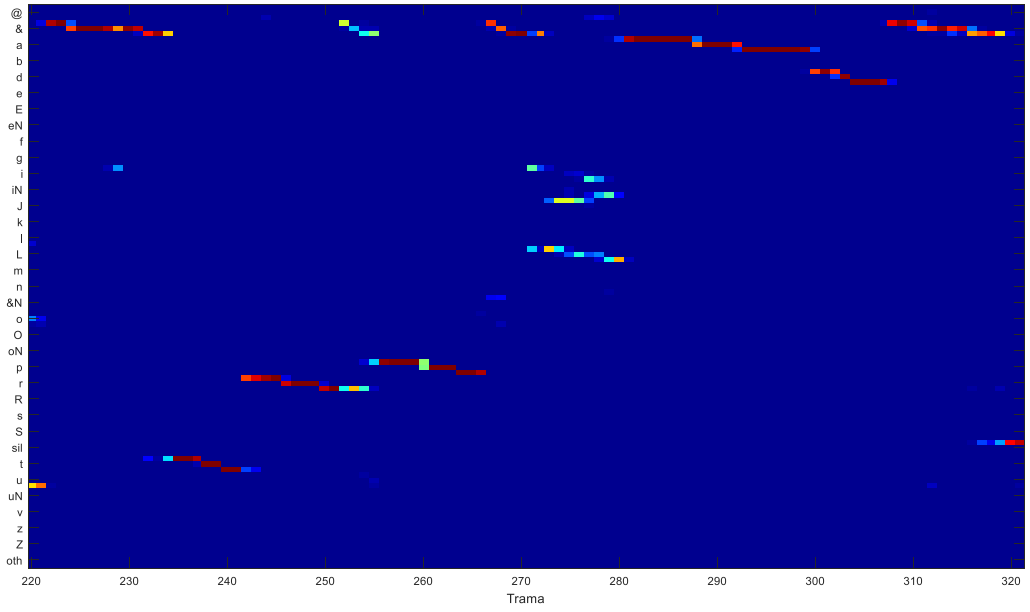

Figura 4.10 - Posteriorgrama do teste 1 (parte da palavra "atrapalhada")

No teste 2, a palavra-chave a verificar é “acontecido” na frase “Como teria acontecido”. Mais uma vez o sistema deteta apenas um pico acima do limar escolhido para teste, *figura 4.11*, o que coincide com o esperado pois o ficheiro não tem nenhuma repetição. Mas na análise do posteriorgrama, *figura 4.12*, o sistema deteta dois erros: que na pronúncia da palavra “acontecido”

[& k oN t @ s i d u] a criança troca o fone [k] pelo [p] e o [s] pelo [f], mas indica que não tem a certeza pois o fone esperado tem uma probabilidade muito próxima do fone com maior probabilidade. Isto é uma falha no sistema que acontece devido à precisão dos modelos De notar que os nossos modelos têm uma precisão de aproximadamente 76% de taxa de acerto dos fones.

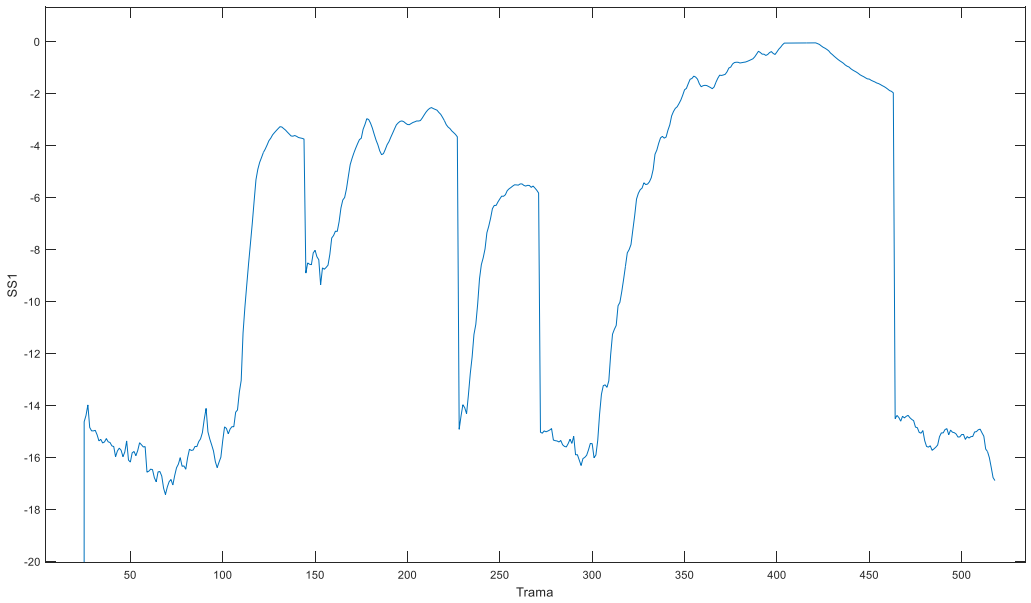

Figura 4.11 - LLR normalizado do teste 2

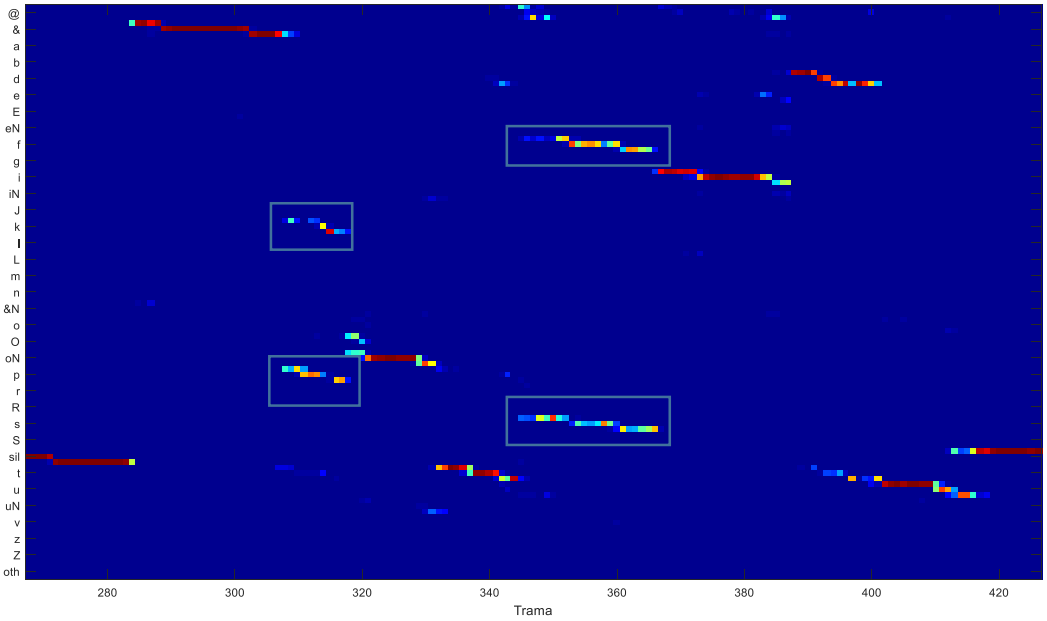

Figura 4.12 - Posteriorgrama do teste 2 (parte da palavra "acontecido")

Como podemos verificar na *figura 4.12*, as cores dos fones previstos e as cores dos fones com maior probabilidade estão com tons muito semelhantes o que indica que as suas probabilidades estão muito próximas.

Depois de verificar a eficácia do sistema em frases corretas, testou-se para frases em que ocorrem erros de pronúncia. Decidiu-se procurar a palavra "conversar" (teste 3) na frase " A Carochinha convidou o João Ratão a entrar, pois tinham muito que conversar e uma data de casamento para marcar", só que neste caso a criança não pronunciou corretamente "conversar". Como se observa na *figura 4.13*, o sistema identifica um pico superior ao limite de decisão porque a falta de um fone correto não vai alterar muito o LLR, por isso é que precisamos de verificar nos posteriorgramas. Esta parte do sistema só nos identifica possíveis partes do discurso que contenham a sequência de fones desejada. Com a análise do posteriorgrama, podemos verificar que a criança trocou o fone [v] por [t] com uma diferença de percentagem suficiente para garantir que a criança pronunciou mal o fone (*figura 4.14*), pois o fone [v] tem um tom de cor muito pouco intenso, o que corresponde com o esperado, pois a criança não o pronunciou.

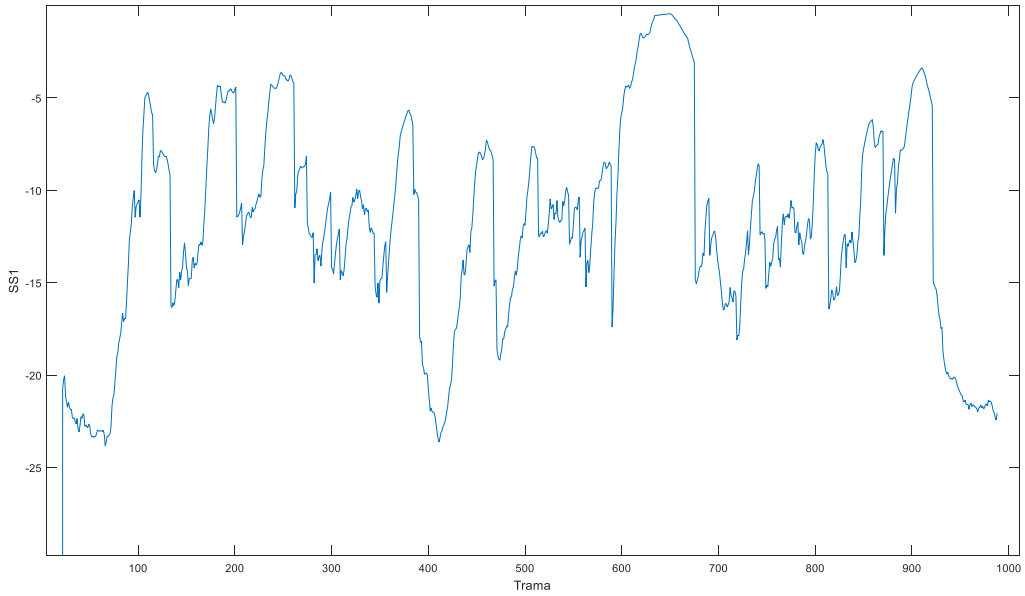

Figura 4.13 - LLR normalizado do teste 3

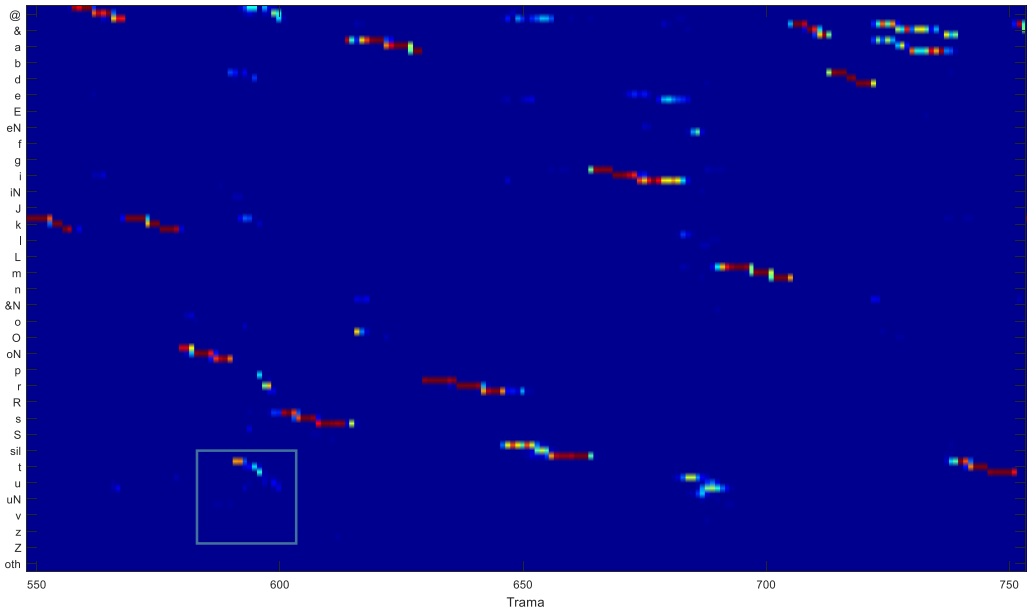

Figura 4.14 - Posteriorgrama do teste 3 (parte da palavra "conversar")

No teste 4 procurou-se a palavra “gostava” e como esperado o sistema encontrou um pico possível correspondente à palavra, *figura 4.15*. Mas no caso da análise dos posteriorgramas o sistema detetou dois erros. Um dos erros está correto pois a criança acentua a palavra gostava no [o] de forma a trocar o fone [o] pelo fone [O]. Quanto ao outro erro o sistema não considera mal apenas diz que o fone [g] não é o fone com maior probabilidade mas que a diferença é muito pouca. Este erro ocorre mais uma vez devido à percentagem de erro dos modelos. No posteriorgrama fica claro a troca do fone [o] pelo [O], *figura 4.16*.

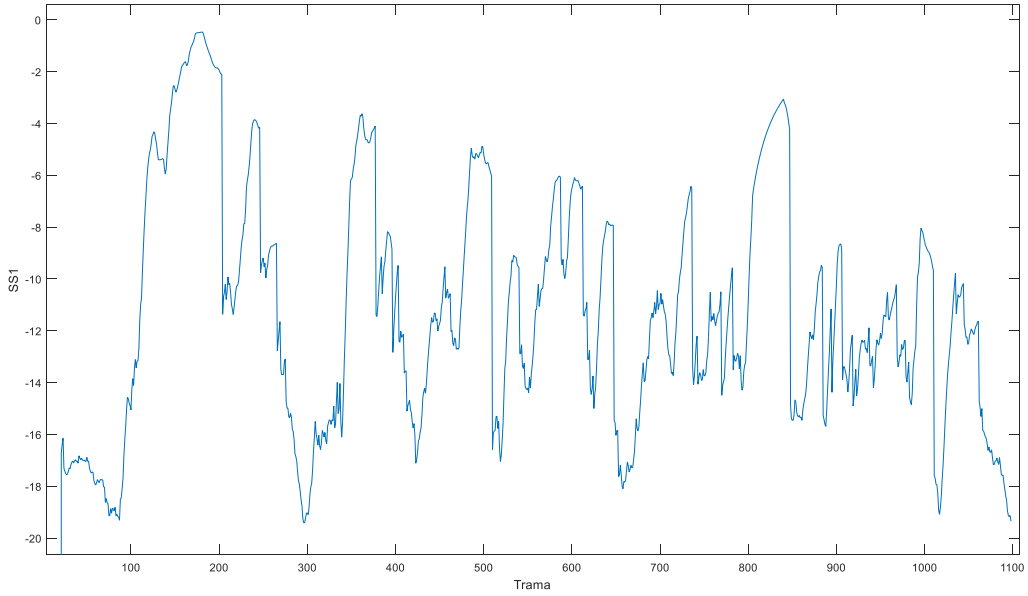

Figura 4.15 - LLR normalizado do teste 4

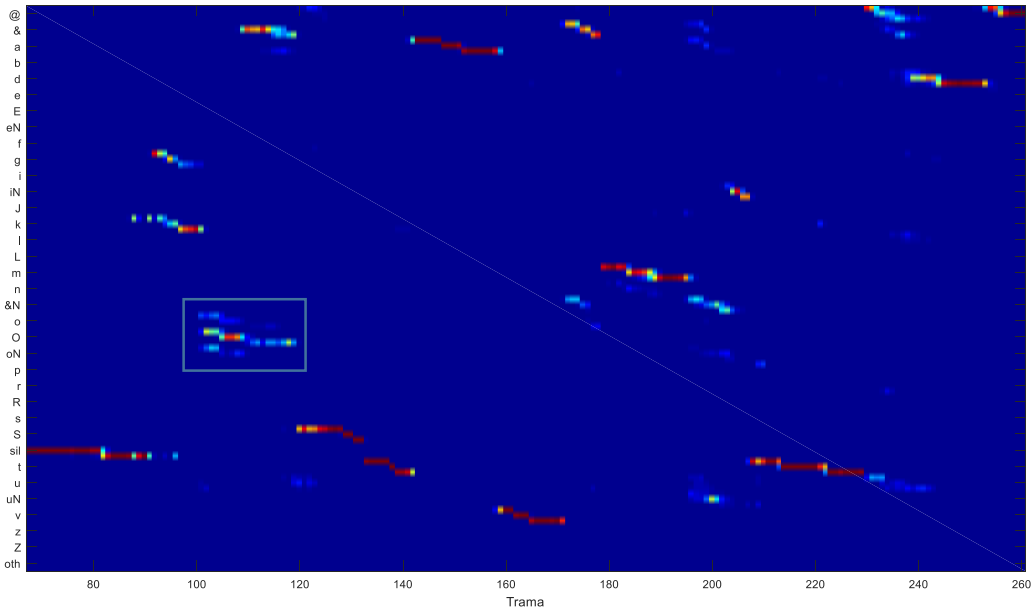

Figura 4.16 - Posteriorgrama do teste 4 (parte da palavra "gostava")

Nos testes anteriores verificámos que o sistema deteta as palavras procuradas mesmo com um ou dois erros de pronúncia. O teste seguinte, teste 5, tem como objetivo verificar se o sistema deteta que a palavra não existe. Para tal, procurou-se pela palavra “ comemoramos” na frase “Em Novembro comemoramos o São Martinho, comendo castanhas cozidas ou assadas” só que neste caso a criança trocou a palavra procurada pela “começaram”. Como se pode observar na *figura*

4.17, não existe nenhum pico que se destaca e que seja superior ao limiar mínimo para que o sistema assuma que existe a palavra.

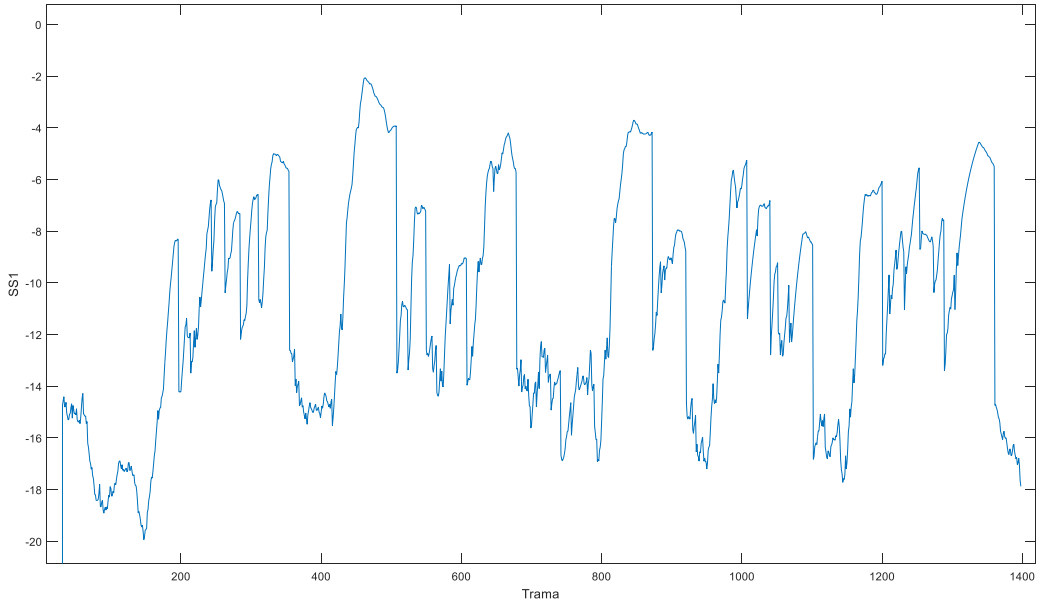

Figura 4.17 - LLR normalizado do teste 5

Outro erro muito comum nos ficheiros da nossa base de dados são as repetições de palavras. Começou-se então por testar a eficiência do sistema a detetar uma repetição - teste 6. Este teste consiste na procura da palavra “encostada”, [iN k u S t a d &] na frase “Mas, de repente, ficou muda de espanto: encostada ao tenro caule de uma planta, descansava uma formiga”, onde a criança repete a palavra. Através da *figura 4.18*, podemos observar que o sistema deteta duas possíveis ocorrências da palavra pesquisada o que indica que é provável que exista repetição. Na análise ao posteriorgrama, o sistema indica que o primeiro pico está relacionado com a palavra “encostada” e que não ocorre nenhum erro. No caso do segundo pico, o sistema deteta novamente que a palavra se encontra nesse pico mas indica dois erros, o primeiro erro mais uma vez ocorre devido a precisão dos modelos pois troca o [iN] por [m], e o segundo erro o sistema indica que a criança pronunciou [&N] em vez de [&]. Este erro também ocorre devido à precisão dos modelos mas também porque o fone [&] quando anterior a alguns fones altera o seu contexto fonético para [&N]. Nestes casos o sistema não indica que o fone está errado mas sim que não é o fone com maior probabilidade e que a diferença é pequena.

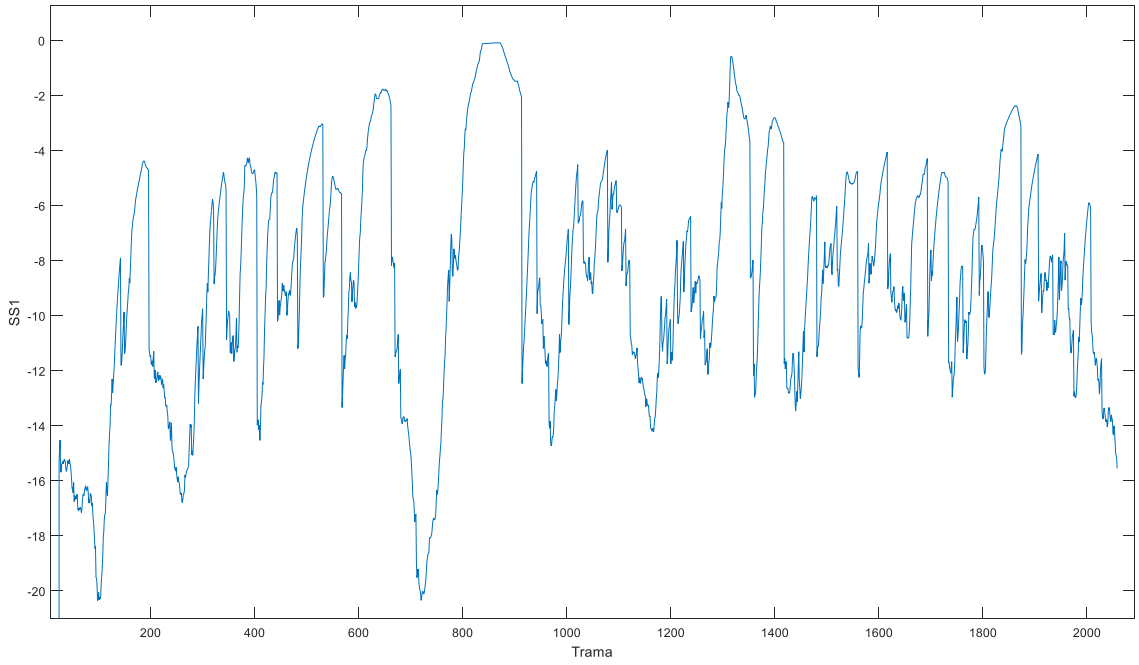

Figura 4.18 - LLR normalizado do teste 6

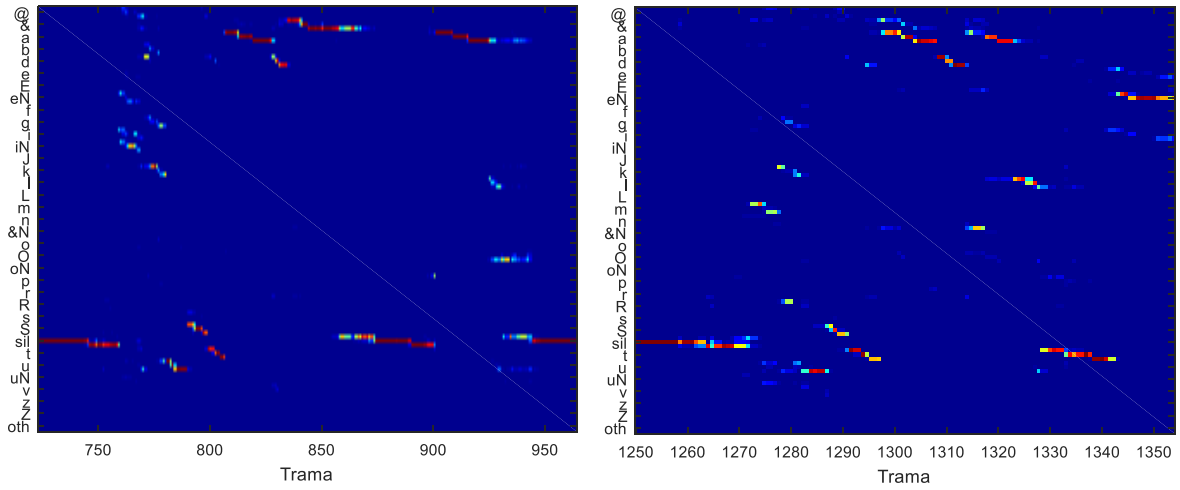

Figura 4.19 - Posteriorgrama do teste 6 (direita: parte da 1ª ocorrência da palavra; esquerda: 2º ocorrência)

Por último, o teste 7 procura a pseudopalavra “chapardas” numa frase em que a criança apenas disse pseudopalavras. Neste caso existe uma repetição da palavra mas com um erro de pronúncia na primeira ocorrência. Através da análise da *figura 4.20* podemos confirmar que existem duas possíveis ocorrências da palavra como corresponde com a realidade. Na análise dos posteriorgramas, *figura 4.21*, o sistema indica que na segunda ocorrência a criança disse a palavra

bem pronunciada, mas que na primeira ocorrência a criança omitiu o fone [r], o que também corresponde com a realidade. O sistema também assume que não tem a certeza sobre o fone [p] pois o fone com maior probabilidade é o fone [&] mas que a diferença é considerada pequena.

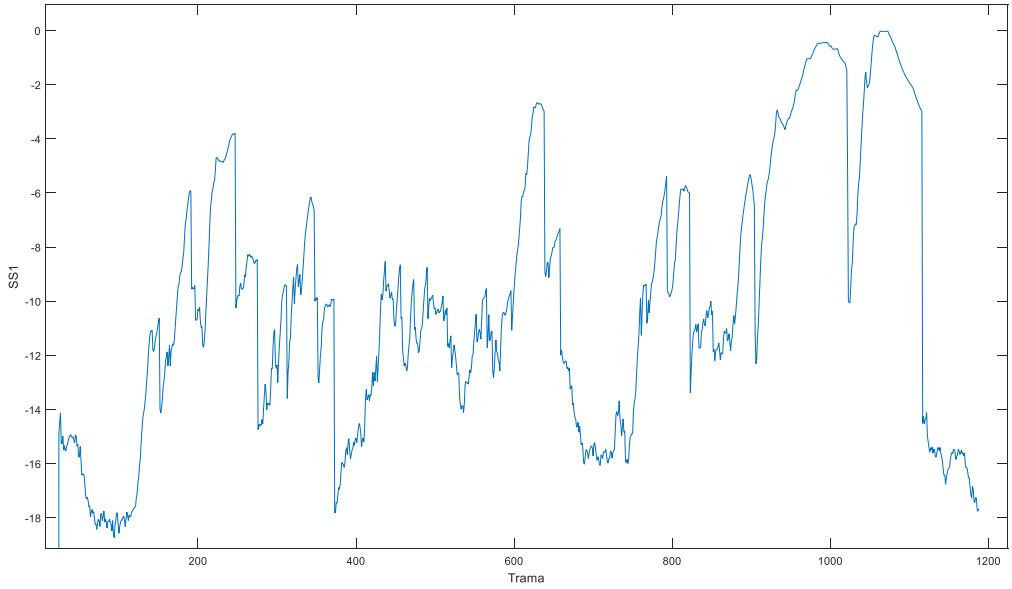

Figura 4.20 - LLR normalizado do teste 7

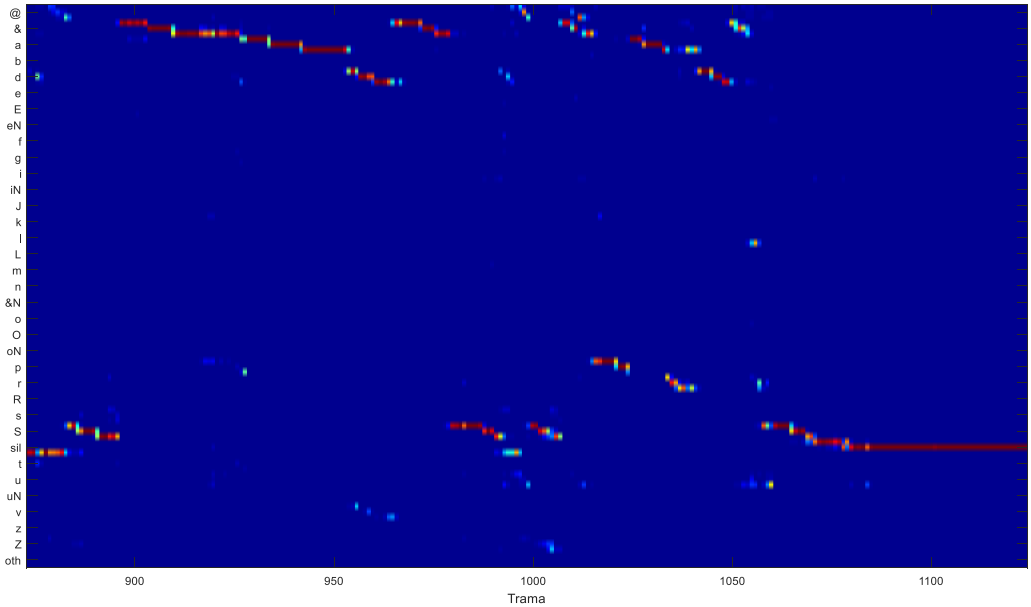

Figura 4.21 - Posteriorgrama do teste 7 (parte de ambas as ocorrências da palavra "chapardas")

Capítulo 5

Conclusão

O principal objetivo desta dissertação consistia em desenvolver técnicas de detenção de eventos acústicos na fala de crianças a ler, de forma a alinhar o texto lido com o áudio gravado. Este objetivo foi atingido. As técnicas desenvolvidas permitem detetar repetições e erros de pronúncia num ficheiro de áudio, com uma boa percentagem de certeza.

Apesar do objetivo ter sido atingido e dos resultados serem satisfatórios, o sistema ainda contém algumas falhas. Como é possível observar nas análises de resultados, o sistema considera alguns erros que não estão corretos devido à semelhança fonética entre fones ou mesmo por falha do sistema, pois os modelos têm uma percentagem de acerto de 76%. Ambos os erros podem ser corrigidos com uma melhoria dos modelos acústicos.

Para trabalho futuro propõe-se melhorar alguns aspetos que proporcionariam uma maior precisão do sistema.

Um aspeto a focar seria na tentativa de aumentar a percentagem de etiquetas manuais da base de dados de forma a obter mais material para treino dos modelos. Este aumento também favorecia o melhoramento do sistema de deteção de eventos acústicos, tornando a escolha de um limiar de decisão mais preciso.

Um outro aspeto importante a desenvolver, é a implementação de um algoritmo que verifique se as partes do áudio definidas como possíveis ocorrências de uma palavra-chave se encontram no mesmo intervalo de tempo do texto de referência.

Concluindo, podemos assumir que o trabalho obteve resultados interessantes, como por exemplo, através do sistema desenvolvido foi possível detetar algumas falhas nas transcrições manuais, assumindo assim que o objetivo principal foi conseguido.

Esta dissertação abrange uma grande variedade de conceitos e de técnicas e é baseada num tema de grande interesse, devido ao aumento da procura da tecnologia como ferramenta de apoio na área da fala. Sendo a leitura das crianças um assunto de elevado interesse, podemos considerar que o desenvolvimento deste tema é bastante atual, desafiador e exigente.

Bibliografia

[1] L. S. Fuchs, D. Fuchs, M. K. Hosp e J. R. Jenkins, "Orl Reading Fluency as a Indicator of Reading Competence: A Theoretical Empirical, and Historical Analysis," *Scientific Studies of Reading,* vol. 5, nº 3, pp. 239-256, 2001.

[2] J. Proença, D. Celorico, S. Candeias, C. Lopes e F. Perdigão, "Children's Reading Aloud Performance: a Database and Automatic Detection of Disfluencies," *Conf. of the International Speech Communication Association - INTERSPEECH,* 2015.

[3] J. Proença, O. Costa, D. Celorico, S. Candeias e F. Perdigão, "Automatic Detection of Disfluencies in Children Reading Aloud Using Task Specific Lattices," *10th Conference on Telecommunications - CONFTELE,* 2015.

[4] Y. Lie, E. Shriberg, A. Stolcke e M. Harper, "Comparing HMM, Maximum Entropy, and Conditional Random Fields for Disfluency Detection," *Proceeedings of the European Conference on Speech Communication and Technology - INTERSPEECH,* 2005.

[5] H. Medeiros, H. Moniz, F. Batista, I. Trancoso e L. Nunes, "Disfluency Detection Based on Prosodic Features for University Lectures," *Proc Annual Conf. of the International Speech Communication Association - INTERSPEECH,* 2013.

[6] M. Black, J. Tepperman, S. Lee, P. Price e S. S. Narayanan, "Automatic Detection and Classification of Disfluent Reading Miscues in Young Children's Speech for the Purpose of Assessment," *INTERSPEECH,* pp. 206-209, 2007.

[7] J. Duchateau, L. Cleuren, H. Van Hamme e P. Ghesquière, "Automatic assessment of children's reading level," *INTERSPEECH,* pp. 1210-1213, 2007.

[8] E. Yilmaz e J. Pelemans, "Automatic Assessment of Children's Reading with the FLaVoR Decoding Using a Phone Confusion Model," *Proceedings Interspeech,* 2014.

[9] X. Li, Y.-C. Ju, L. Deng e A. Acero, "Efficient and Robust Language Modeling in an Automatic Children's Reading Tutor System," *IEEE International Conference on Acoustics, Speech and Signal Processing,* 2007.

[10] “Projeto LetsRead,” [Online]. Available: https://www.it.pt/Projects/Index/1938/. [Acedido em 2015].

[11] H. C. Buescu, J. Morais, M. R. Rocha e V. F. Magalhães, “Programa e Metas Curriculares de Português do Ensino Básico,” *Ministério da Educação e Ciência,* 2015.

[12] L. R. Rabiner e B. H. Juang, “An Introduction to Hidden Markov Models,” *IEEE ASSp Magazine,* pp. 4-16, 1986.

[13] J. Yamagishi, An Introduction to HMM-Based Speech Synthesis, 2006.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. (. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev e P. Woodland, The HTK Book v3.4.1, 2009.

[15] A. Veiga, C. Lopes, L. Sá e F. Perdigão, “Acoustic Similarity Scores for Keyword Spotting,” *11th International Conference - PROPOR,* pp. 48-58, 2014.

[16] A. Hämäläinen, F. M. Pinto, S. Rodrigues, A. Júdice, S. M. Silva, A. Calado e M. S. Dias, “A Multimodal Educational Game for 3-10-Year-Old Children: Collecting and Automatically Recognising European Portuguese Children’s Speech,” *Proceedings of Workshop on Speech and Language Technology in Education,* 2013.

[17] Ö. G. Saracoglu e H. Altural, “Color Regeneration from Reflective Color Sensor Using an Artificial Intelligent Technique,” *Sensores,* vol. 10, nº 9, pp. 8363-8374, 2010.

[18] Luis Miguel Bagagem Castela, “Pesquisa por Áudio. Dissertação de Mestrado,” Faculdade de Ciências e Tecnologia – Universidade de Coimbra, Coimbra, 2015.

[19] H. Boulard e N. Morgan, Connectionist speech recognition: A hybrid approach., Academic Publishers, 1994.

Anexo A

Tabela de fonemas

Tabela A.1 - Tabela de fonemas em SPL-IT e SAMPA

Tipo	SPL-IT	SAMPA_UC	SAMPA	Exemplo	Transcrição Fonética
Consoantes Plosivas	p	p	p	pai	p a i
	b	b	b	barco	b a r k u
	t	t	t	tenho	t e J u
	d	d	d	doce	d o s @
	k	k	k	com	k oN
	g	g	g	grande	g r &N d &
Consoantes Fricativas	f	f	f	falo	f a l u
	v	v	v	verde	v e r d &
	s	s	s	céu	s E w
	z	z	z	casa	k a z &
	S	S	S	chapéu	S & p E u
	Z	Z	Z	jóia	Z O i &
Consoantes Nasais	m	m	m	mar	m a r
	n	n	n	nada	n a d &
	J	J	J	vinho	v i J u
Consoantes Líquidas	l	l	l	lanche	l aN S @
	L	L	L	trabalho	t r & b a L u
	r	r	r	caro	k a r u
	R	R	R	rua	R u &
Vogais	i	i	i	lápis	l a p i S
	e	e	e	fazer	f & z e r
	E	E	E	belo	b E l u
	a	a	a	falo	f a l u
	&	&	6	cama	k & m &
	O	O	O	roda	R O d a
	o	o	o	lobo	L o b u
	u	u	u	futuro	f u t u r u
	@	@	@	felizes	f @ l i z @ s
	iN	ï	i~	fim	f iN
	eN	ë	e~	emprego	eN p r e g u
	&N	ä	6~	irmã	i r m aN
	oN	ö	o~	bom	b oN
	uN	ü	u~	um	uN