6
Condições Experimentais e Resultados

Para avaliar o desempenho das técnicas descritas nos capítulos anteriores, é fundamental realizar experimentos práticos. Esta seção explicará em detalhes como os testes foram realizados.

6.1
Banco de Dados para Treino e Teste

Para treinar e testar um reconhecedor de voz contínua independente de locutor, é preciso ter uma quantidade grande de frases gravadas por muitas pessoas. Além disso, o sistema precisa ser testado em cenários com diferentes tipos de ruído. Por isso, esta dissertação utilizou dois bancos de dados comerciais contendo arquivos de sons. Esses bancos serão detalhados a seguir.

6.1.1.
Banco de Dados de Voz

Um banco de dados de voz contém gravações de pessoas pronunciando palavras ou frases variadas. Dentre as opções existentes no mercado, foi escolhido a base TIMIT [33], cuja sigla vem de *Texas Instrument* (TI) e *Massachusetts Institute of Technology* (MIT). Ele abrange os diversos sotaques do inglês americano para ambos os sexos.

TIMIT foi criado especialmente para experimentos de reconhecimento de voz contínua independente de locutor. Ele possui gravações de 630 pessoas pronunciando 10 frases cada, resultando em 4620 sentenças para treino e 1680 para teste – a Figura 56 mostra alguns exemplos. A taxa de amostragem das gravações é de 16 kHz, com apenas um canal (mono) e valores de 16 bits.

Locutor 1:	she had your dark suit in greasy wash water all year don't ask me to carry an oily rag like that even then if she took one step forward he could catch her (...) critical equipment needs proper maintenance tim takes sheila to see movies twice a week	} 10 frases
Locutor 2:	she had your dark suit in greasy wash water all year don't ask me to carry an oily rag like that receiving no answer they set the fire (...) a huge power outage rarely occurs academic aptitude guarantees your diploma	} 10 frases

(...)

Figura 56:Exemplos de frases da base TIMIT.

A Figura 56 anterior mostra que algumas frases se repetem entre os locutores. Mais precisamente, existem apenas 2342 sentenças distintas. Além de guiar o treinamento e o teste dos HMMs, elas serão usadas para criar o modelo de linguagem.

Para formar as frases, foram usadas 6234 palavras. Cada uma delas é associada a uma sequência de fones, num dicionário de pronúncia como o da Figura 57. Conforme visto em 2.2, isso será usado para treinar HMMs representando monofones, bifones e trifones.

abbreviate	ax b r iy v iy ey t	} 6234 palavras		aa	} 45 fones
abdomen	ae b d ax m ax n			ae	
abides	ax b ay d z			ah	
(...)				(...)	
zoologist	z ow aa l ax jh ix s t			z	
zoos	z uw z			zh	

Figura 57:Amostra do dicionário de palavras da base TIMIT.

Cabe ressaltar que a base continha mais do que 45 fones originalmente. O autor substituiu alguns deles por outros similares, para diminuir a quantidade total e melhorar assim o desempenho do reconhecedor.

6.1.2. Banco de Dados de Ruído

Apesar de ser bem completa, a base TIMIT tem apenas amostras limpas de voz. O objetivo desta dissertação é testar métodos que melhorem o reconhecimento em presença de ruído. Para isso, serão necessários sinais

corrompidos de diversas maneiras (por exemplo, com um chiado bem baixo ou um falatório bem alto).

Para obter as amostras corrompidas de teste, basta tomar as amostras limpas de voz e adicionar um sinal de ruído sobre ela. Dependendo da intensidade desse ruído, a voz será corrompida com uma intensidade maior ou menor. Um meio de medir essa intensidade é através da razão sinal-ruído (RSR) [34]: a razão entre a energia do sinal de voz $s(n)$ e a energia do ruído $r(n)$ é convertida para a escala de decibéis com a expressão

$$RSR = 10\log\left(\frac{\sum_{n=1}^{N} s^2(n)}{\sum_{n=1}^{N} r^2(n)}\right) \tag{45}$$

Este trabalho selecionou sinais de ruído $r(n)$ a partir de uma segunda base de dados: a NOISEX-92 [35]. Ela contém arquivos de som variados, como falatório, trânsito e ruído branco. Trechos aleatórios desses sinais foram adicionados às amostras de teste, com razões sinal-ruído variando de 20 dB a 5 dB. O sinais corrompidos e os limpos foram usados para obter os resultados listados em 6.4.

Nota-se que, ao contrário do teste, o treinamento foi realizado apenas com sinais limpos.

6.2 Ferramentas Computacionais

O sistema de reconhecimento foi implementado com a ferramenta HTK (HMM *Tool Kit*) [36][37]. Trata-se de um software gratuito, aberto e multiplataforma. Ele possui um conjunto de programas que são chamados por linhas de comando no terminal do sistema – alguns exemplos podem ser vistos na Figura 58. Com a ajuda deles, é possível realizar uma série de tarefas básicas, desde a extração de atributos até a listagem dos resultados.

```
HCopy -C configuration.txt -S files.txt
```

extração de atributos MFCC dos arquivos listados em "files.txt", usando os parâmetros descritos no arquivo "configuration.txt"

```
HResults -I expectedWords.txt triphones.txt recognizedWords.txt
```

calcula o resultado do reconhecimento comparando as palavras esperadas com as reconhecidas

Figura 58:Exemplos de comandos da ferramenta HTK.

Usando apenas o HTK, seria necessário repetir um a um os comandos em cada experimento, além de criar e mover diversos arquivos manualmente. Outro problema da ferramenta é que ela não extrai coeficientes SSCH e PNCC, e nem realiza nenhum tipo de *denoising*. Para resolver essas deficiências, um segundo software foi escolhido: Matlab.

Matlab [38] é um programa pago, fechado e multiplataforma, voltado para diversas aplicações de engenharia. Através de uma linguagem de programação própria, é possível manipular grandes quantidades de dados numéricos com poucos comandos. Ele também traz suporte a números complexos e uma série de funções matemáticas (incluindo transformada *wavelet* e redes neurais). A Figura 59 apresenta exemplos da interface do software.

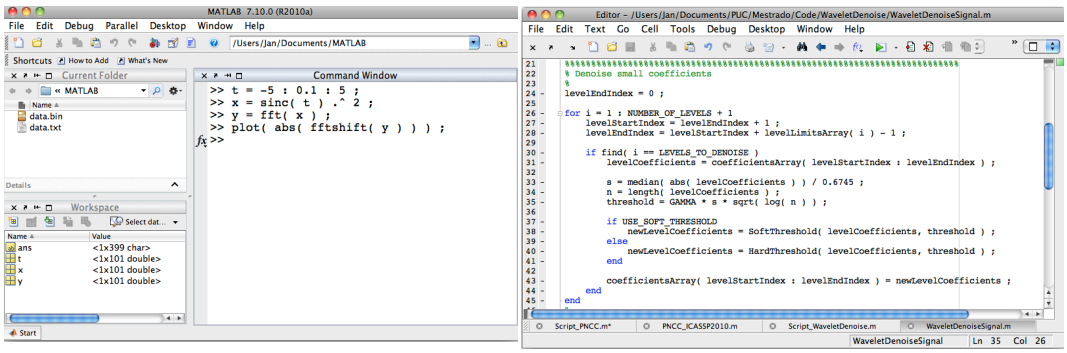

Figura 59:Exemplos de janelas do programa Matlab.

Portanto, a execução dos treinamentos e testes foi toda coordenada por comandos no Matlab, que gerenciou os arquivos de dados e invocou os programas do HTK, além de realizar a extração dos SSCH e PNCC e aplicar os métodos de *denoising*.

Todo o código criado no trabalho é gratuito, aberto e multiplataforma, estando disponível em [39]. O autor encoraja que pesquisadores da área façam bom uso do material e contribuam para seu aperfeiçoamento.

6.3
Parâmetros Escolhidos

Nos capítulos 2, 3, 4 e 5, muitas expressões matemáticas possuem parâmetros arbitrários – tais como o número de estados dos HMMs ou a função mãe da transformada wavelet. Esses parâmetros foram ajustados experimentalmente e estão descritos nas subseções a seguir.

6.3.1.
Parâmetros do Reconhecimento de Voz

Os HMMs foram construídos com $N = 5$ estados (vide seção 2.1), na configuração mostrada na Figura 60.

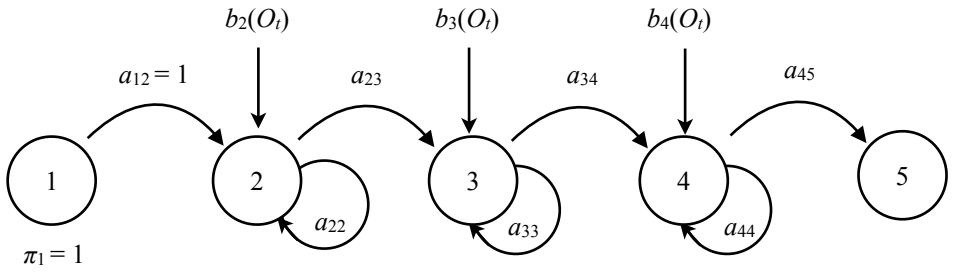

Figura 60:Configuração dos HMMs utilizados.

Inicialmente, foram gerados modelos de monofones. A partir deles, os modelos de trifones foram estimados com a configuração intra-palavra (vide seção 2.2).

Nos estados 2, 3 e 4 dos HMMs, foi criada uma densidade de probabilidade com apenas uma gaussiana inicialmente. Em seguida, os modelos foram reestimados acrescentando novas densidades, uma a uma, até alcançar o total de $M = 8$ gaussianas (vide seção 2.4).

O peso do modelo de linguagem L (vide seção 2.3) foi ajustado para 15. A estatística de ocorrência de palavras foi estimada a partir das frases do banco de dados TIMIT.

6.3.2.
Parâmetros da Extração de Atributos

Durante a extração de atributos, os sinais de áudio tiveram a taxa de amostragem reduzida para 8 kHz, buscando simular a faixa do sistema telefônico.

O filtro de pré-ênfase usou $\alpha = 0.97$ (vide seção 3.1). Os quadros do sinal tiveram duração de 25 ms, com um passo de 10 ms entre um e outro (vide seção 3.2). Coeficientes delta e de aceleração foram calculados com $\Delta = 2$ (vide seção 3.6).

Além disso, cada um dos métodos da seção 3.4 usou os seguintes parâmetros.

- **MFCC**: $B = 26$ bandas, somente os 12 primeiros valores da DCT foram considerados e coeficientes delta e de aceleração foram incluídos.
- **SSCH**: $B = 60$ bandas, $I = 15$ intervalos, todos os 15 valores da DCT foram considerados e apenas coeficientes delta foram incluídos (os de aceleração pioravam o desempenho).
- **PNCC**: $B = 40$ bandas com filtros de ordem $n = 1$, expoente $a_0 = 1/15$, somente os 20 primeiros valores da DCT foram considerados e apenas coeficientes delta foram incluídos (os de aceleração não afetavam o desempenho).

6.3.3. Parâmetros do Wavelet Denoising

Para a transformada wavelet apresentada na seção 4.2, escolheu-se a função mãe Daubechie 10. Seus valores $W(a,b)$ foram obtidos para 5 níveis, além do nível de aproximação.

O valor de γ no cálculo do limiar (vide seção 4.4) foi ajustado para 0.2 ou 0.1, de acordo com a intensidade do ruído. Mais detalhes sobre isso se encontram na subseção 6.4.2.

6.3.4. Parâmetros do Feature Denoising

Para o feature denoising apresentado no capítulo 5, foram usadas redes neurais do tipo *feedforward*, todas com 1 camada escondida contendo 10 neurônios. A função de ativação foi a tansig, definida em (44).

Um conjunto de redes foi gerado separadamente para cada um dos três métodos de extração (MFCC, SSCH e PNCC). Para ajustar os pesos dos neurônios, foram escolhidas 100 amostras de voz do banco de dados TIMIT

(dentre suas 4620 amostras de treinamento). A partir delas, amostras corrompidas foram criadas adicionando ruído branco, com razões sinal-ruído de 20 dB e 10 dB. Seus atributos foram extraídos e usados conforme o treinamento descrito na seção 5.2.

6.4
Testes e Resultados

O desempenho de um reconhecedor de voz é dado pela taxa de acerto das palavras nas frases de teste. Esse valor é o número de palavras esperadas subtraído dos erros, e depois normalizado pelo número de palavras esperadas [37, p. 184]. Matematicamente falando, tem-se

$$R = \frac{\text{total} - \text{erros}}{\text{total}} = \frac{N - (S + D + I)}{N}$$
$$R(\%) = 100\,\frac{N - (S + D + I)}{N} \tag{46}$$

onde N é o número de palavras esperadas no teste, S é o número de palavras substituídas, D é o número de palavras deletadas e I é o número de palavras inseridas. A Figura 61 mostra um exemplo desse cálculo.

```
frase original:    O LÍDER MANDOU CHAMÁ-LO   SEGUNDO A IMPRENSA.
frase reconhecida: O LÍDER MANDOU CHAMÁ-LO O SEGUNDA   IMPRENSA.

TOTAL:        N = 7 palavras (O LÍDER MANDOU CHAMÁ-LO SEGUNDO A
IMPRENSA)
SUBSTITUÍDAS: S = 1 palavra  (SEGUNDO → SEGUNDA)
DELETADAS:    D = 1 palavra  (A)
INSERIDAS:    I = 1 palavra  (O)

     TAXA DE ACERTO: R(%) = 100 * ( 7 - 1 - 1 - 1 ) / 7 = 57.14%
```

Figura 61:Exemplo do cálculo da taxa de acerto para uma frase.

A seguir, a taxa de acerto foi usada para avaliar o desempenho das técnicas apresentadas nos capítulos anteriores.

6.4.1.
Teste dos Métodos de Extração de Atributos

O primeiro teste foi aplicado sem as técnicas de *wavelet denoising* e *feature denoising*. O objetivo era comparar apenas o desempenho dos três métodos de extração de atributos: MFCC, SSCH e PNCC.

Na Tabela 1, encontram-se as taxas de acerto para amostras corrompidas com ruído branco (chiado) e falatório. Como o falatório prejudicava menos o desempenho que o ruído branco, seus testes foram realizados com razões sinal-ruído menores (sinais mais corrompidos). Desse modo, a queda das taxas ficou mais similar nos dois casos.

Tabela 1: Taxas de acerto utilizando apenas extração de atributos

		Ruído Branco			Falatório		
	limpo	**20 dB**	**15 dB**	**10 dB**	**15 dB**	**10 dB**	**5 dB**
MFCC	84.19%	69.51%	45.96%	18.20%	72.13%	43.59%	11.60%
SSCH	75.88%	68.49%	51.99%	29.92%	65.76%	48.24%	16.15%
PNCC	88.40%	79.98%	72.81%	64.96%	79.18%	69.17%	39.59%

Nota-se que método MFCC teve uma queda de desempenho muito grande com a adição de ruído, comprovando o comportamento visto em [3][4][5]. Os SSCH tiveram taxas melhores que os MFCC, mas apenas para os casos de menor razão sinal-ruído. Já os PNCC foram claramente os mais robustos, com resultados melhores em todos os cenários analisados.

Cabe ressaltar novamente que não existia na literatura uma comparação direta entre os três métodos.

6.4.2. Teste do Wavelet Denoising

O segundo teste usou as mesmas condições do primeiro, mas aplicando o *wavelet denoising* antes da extração dos atributos. O objetivo é constatar se há uma melhora no resultado de cada cenário.

As novas taxas de acerto seguem na Tabela 2. Nota-se que o parâmetro γ da seção 4.3 teve que ser ajustado experimentalmente de acordo com os níveis de ruído, buscando melhores resultados. Em sistemas reais, esses níveis poderiam ser estimados analisando o som do ambiente antes de o locutor começar a falar.

Tabela 2: Taxas de acerto utilizando wavelet denoising e extração de atributos

		Ruído Branco			Falatório		
	limpo	**20 dB**	**15 dB**	**10 dB**	**15 dB**	**10 dB**	**5 dB**
MFCC	80.31%	76.79%	62.34%	34.36%	76.45%	52.56%	21.16%
SSCH	73.95%	70.31%	64.39%	38.57%	70.65%	56.09%	21.39%
PNCC	84.19%	81.23%	73.72%	65.63%	79.52%	70.31%	44.14%

Comparando a Tabela 2 com a Tabela 1, percebe-se que o denoising ajudou bastante o desempenho dos MFCC, que antes eram muito prejudicados pelo ruído. A melhora com os SSCH foi menor, pois já existia uma robustez na extração de atributos. Já os PNCC praticamente não foram afetados, provavelmente por já contarem com uma técnica de remoção de ruído (vide subseção 3.4.3).

Em todos os métodos, o cenário limpo é um pouco prejudicado. Isso era esperado, pois a transformada wavelet de um sinal limpo contém apenas dados sobre a voz. Ou seja, qualquer *denoising* ali elimina informações necessárias para o reconhecimento.

Assim como em 6.4.1, a comparação desses resultados é inédita na literatura. Inclusive, não foi encontrada sequer uma referência sobre *wavelet denoising* com SSCH e PNCC.

6.4.3. Teste do Feature Denoising

O terceiro teste usou as mesmas condições do primeiro, mas aplicando o *feature denoising* após a extração dos atributos. Novamente, o objetivo é constatar se há uma melhora no resultado de cada cenário. As novas taxas de acerto seguem na Tabela 3.

Tabela 3: Taxas de acerto utilizando extração de atributos e feature denoising

		Ruído Branco			Falatório		
	limpo	**20 dB**	**15 dB**	**10 dB**	**15 dB**	**10 dB**	**5 dB**
MFCC	58.02%	76.34%	68.71%	49.72%	63.25%	50.28%	24.35%
SSCH	63.59%	68.83%	62.34%	46.53%	56.88%	49.49%	30.6%
PNCC	81.11%	80.32%	76.34%	71.56%	79.86%	74.06%	52.79%

Comparando a Tabela 3 com a Tabela 1, nota-se que houve uma melhora em quase todos os cenários (exceto no limpo de no falatório 15 dB). Já em comparação com a Tabela 2, a melhora foi menor. Houve um aumento na maioria das taxas dos PNCC, com ganhos significativos nos cenários mais ruidosos. Já os SSCH e MFCC tiveram resultados inferiores em quase todos os casos, exceto com ruído branco 10 dB e falatório 5 dB. Entretanto, como o PNCC é o método com as taxas mais altas, pode-se dizer que o *feature denoising* se mostrou superior ao *wavelet denoising*.

Por fim, é interessante notar que as redes neurais conseguiram atenuar o efeito do falatório, ainda que seu treinamento tenha recebido apenas ruído branco. Além disso, ao contrário do teste anterior, nenhum parâmetro teve que ser ajustado manualmente de acordo com o cenário. Isso aponta uma boa generalização do *feature denoising*, que pode vir a ser eficaz para outros tipos de ruído com potências variadas.

6.4.4. Teste do Sistema Completo

O último teste aplicou em sequência o *wavelet denoising*, extração de atributos e *feature denoising*. O objetivo é averiguar se a combinação de todas as técnicas traz mais benefícios do que a combinação parcial dos testes anteriores. Os resultados seguem na Tabela 4.

Tabela 4: Taxas de acerto utilizando wavelet denoising, extração de atributos e feature denoising

		Ruído Branco			Falatório		
	limpo	20 dB	15 dB	10 dB	15 dB	10 dB	5 dB
MFCC	58.02%	76.34%	68.71%	49.72%	63.25%	50.28%	24.35%
SSCH	63.59%	68.83%	62.34%	46.53%	56.88%	49.49%	30.6%
PNCC	81.11%	80.32%	76.34%	71.56%	79.86%	74.06%	52.79%

Infelizmente, os resultados com as três técnicas encadeadas apresentaram melhora apenas o cenário de ruído branco 10 dB para o método SSCH. A explicação mais provável é que o wavelet denoising removeu dados importantes para a rede sobre o sinal de voz.