UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PESQUISA

PROGRAMA DE INICIAÇÃO CIENTÍFICA VOLUNTÁRIA – PICVOL

Aperfeiçoamento do Sistema de Conversão Português-LIBRAS

Área do conhecimento:Sistemas de Telecomunicações
Subárea do conhecimento:Reconhecimento de Fala
Especialidade do conhecimento:Tradutor Português-LIBRAS utilizando reconhecimento de fala

Relatório Final

PICVOL

Sumário

1 Resumo

A comunicação interpessoal se baseia na habilidade da troca de informações entre dois ou mais indivíduos. Apesar de ser bastante comum no âmbito da modalidade oral, existe uma precariedade quando se envolve as modalidades oral e gesto-visual. Visto a necessidade de comunicação interpessoal independente da modalidade usada, o presente trabalho propõe o aperfeiçoamento de um sistema de conversão Português-LIBRAS. O desenvolvimento desta ferramenta tem por objetivo estreitar as barreiras de comunicação entre surdos e ouvintes e auxiliar na comunicação interpessoal.

2 Palavras-chave

Tradutores de Idiomas, Reconhecimento de Fala, Modelo Oculto de Markov.

3 Introdução

O sistema de conversão entre idiomas é responsável por avaliar e realizar o mapeamento dos sinais de fala para os sinais em LIBRAS (Língua Brasileira de Sinais) correspondentes por meio de um algoritmo de busca. Esse algoritmo deve ser capaz de identificar a palavra gerada do reconhecimento de fala, classificá-la de acordo com o dicionário em LIBRAS e buscar o sinal correspondente àquela palavra, caso contrário, o sistema optará pela classificação mais simples: letra a letra.

Para o reconhecimento de fala, o Modelo Oculto de Markov (HMM) apresenta-se como principal técnica para este trabalho. Dessa maneira, é possível realizar o treinamento do sistema e obter boas taxas de reconhecimento que se refletem na eficácia do conversor.

3.1 Tradutores de Idiomas

Seja na ficção ou na vida real, os tradutores de línguas orais desempenham um importante papel nos sistemas de comunicação. Ferramentas como essas, são constantemente aprimoradas para auxiliar as pessoas.

Travis *Touch*, Smark *Translator*, *Muama Enence*, Microsoft *Translator*, Ili. Esses tradutores portáteis de idiomas têm desempenhado um importante papel no desenvolvimento da tecnologia dos reconhecedores de fala.

Figura 1: Tradutores portáteis de idiomas [1].

No entanto, o mesmo não poder ser afirmado aos tradutores de línguas oral e gesto-visual visto que as ferramentas para essa modalidade não acompanham o desenvolvimento da tecnologia dos tradutores orais convencionais. O que estabelece um desafio na inclusão da língua de sinais aos tradutores tradicionais.

O último censo realizado pelo IBGE (2010), estima que há cerca de 9,7 milhões de pessoas com algum grau de deficiência auditiva. Dentre esses, há pouco mais de 2 milhões de pessoas com deficiência auditiva severa; caracterizada pela perda auditiva na faixa entre 70 e 90 decibéis (dB) [2].

A LIBRAS é o segundo idioma oficial do Brasil de acordo com a Lei 10.436 de 24 de abril de 2002. No entanto, a naturalização como segunda língua oficial é ínfima e com pouco impacto social.

Diante do exposto, o presente trabalho propõe o aperfeiçoamento de um sistema de acessibilidade à pessoas surdas para diminuir as fronteiras existentes entre surdos e leigos com a LIBRAS.

3.2 Processo de Produção da Fala

A fala se origina pela liberação do ar dos pulmões para o trato vocal. O trato vocal é formado pelas cavidades e órgãos articuladores que começam na abertura entre as dobras vocais (glote) e encerra nos lábios. Sua estrutura tubular pela qual passa o fluxo de ar originado dos pulmões é modulado na glote. A glote é responsável por modular o espectro de frequência da onda sonora e realizar constrições para a geração de sons. Dessa forma, pode-se associar as dobras vocais à fonte sonora e o trato vocal, ao filtro; cuja Figura 2 ilustra [3] [4] [5] [6].

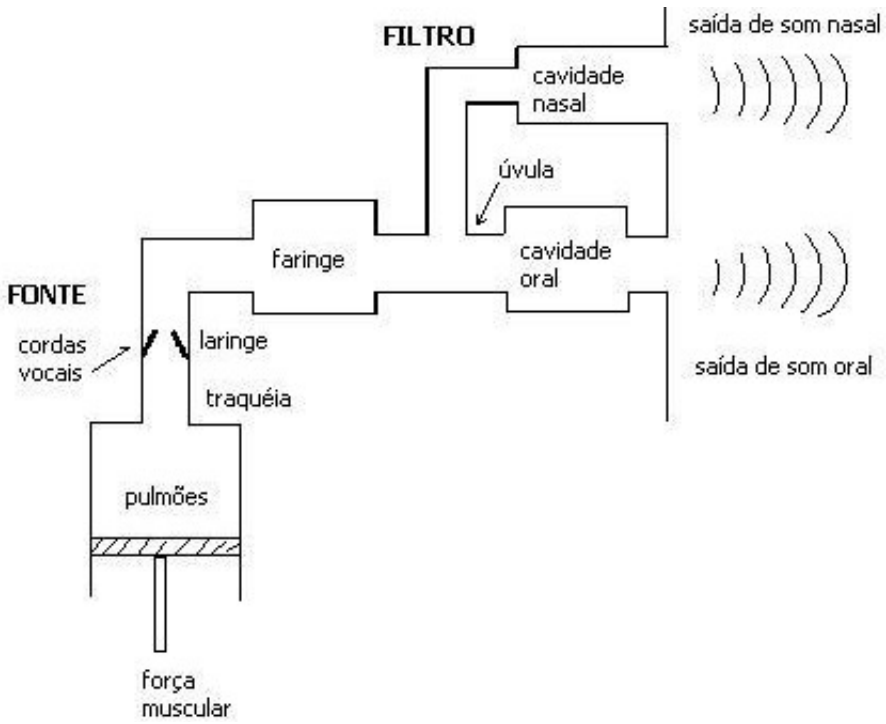

Figura 2: Diagrama de Blocos da Produção da Fala [3]

Os sinais de fala constituem-se de uma sequência ordenada de sons regida por regras de linguagem que dão significado a informação transmitida. Sua produção fisiológica se dá pelo aparelho fonador que constitui-se dos sistemas respiratório, fonatório e articulatório [7] [4] [6] [8].

O emprego da fala na comunicação humana baseia-se em uma sintaxe de léxicos e nomes de um vocabulário. Define-se léxico como uma biblioteca de palavras disponíveis numa língua. A combinação de vogais e consoantes gera os fonemas [9] [10] [6].

A voz é uma onda de pressão acústica que se origina a partir de movimentos voluntários dos órgãos vocais humanos. Sendo que a voz é o som da fala [8].

Durante o processo de fonação, o ar armazenado nos pulmões é expelido passando pelas pregas vocais, que por sua vez podem vibrar ou não, constituindo dessa forma, a fonte de excitação para a produção de alguns sons. No trato vocal, esse fluxo de ar sofre influência das estruturas ressonantes, produzindo os diversos sons da fala [8] [5].

Em suma, a produção da fala é composta por três partes fundamentais: fonte de excitação, trato vocal e radiação [8].

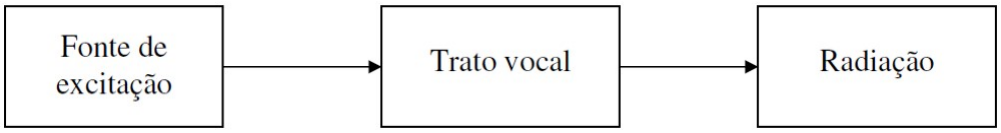

Figura 3: Diagrama das partes fundamentais da produção da fala [8].

A fonte de excitação é responsável por excitar (alimentar) o trato vocal, que por sua vez atua como um filtro acústico, variante no tempo, que modela o sinal de fala através da ação de seus articuladores. Por último, esse sinal é irradiado pela boca ou nariz, dependendo do tipo de som em consideração [8] [6].

3.3 Fonética e Fonologia

A fonética é responsável por estudar os sons da fala nos aspectos acústico e fisiológico e analisar as formas de pronúncia das vogais e consoantes; principalmente para descrição, classificação e transcrição [11].

A fonologia preocupa-se com o estudo dos fonemas e seus padrões em uma determinada linguagem. Os fonemas podem ser vistos como uma representação das diferentes partes da fala cujo objetivo é a combinação dos sons que exprimam significado [12] [13] [14].

Os fonemas representam classes de sons e a realização acústica de um fonema chama-se fone. O fone é sensível à localização que se encontra numa palavra. Isso é denominado de contexto fonético e é causado por um fenômeno chamado de coarticulação. Essas versões modificadas de um fone causadas pela coarticulação são chamadas de alofones. Para um reconhecimento de fala robusto é importante utilizar dessa natureza dos fonemas dependentes de contexto para a criação de modelos detalhados de fonemas [12] [14].

3.4 Parâmetros do Reconhecimento de Fala

Os sistemas de reconhecimento de fala apresentam uma natureza interdisciplinar. Envolve áreas tais como processamento de sinais, reconhecimento de padrões, inteligência artificial, linguística, teoria das comunicações, fonética articulatória e acústica [15].

De modo geral, os sistemas de reconhecimento de fala precisam estar aptos a funcionar sob condições de ruído ambiente, o que exige robustez ao sistema. Existem alguns parâmetros aos quais são classificados, tais como [16] [17]:

• Modo de pronúncia - O sistema pode requerer que o locutor realize pequenas pausas entre as palavras ou pode ser de fala contínua que não apresenta este inconveniente [16] [17] [11].

• Estilo de pronúncia - A fala espontânea possui mais coarticulações por ser mais relaxada e portanto mais difícil de reconhecer quando comparada ao estilo de leitura [16] [11] [15].

• Treinamento - Podem ser classificados como dependentes ou independentes de locutor. Neste não há necessidade de uma fase de treinamento pois já foi feita previamente; enquanto que aquele precisa que o sistema seja treinado pelo usuário [16] [17] [11].

• Vocabulário - O grau de sucesso do reconhecimento de fala é inversamente proporcional ao crescimento do vocabulário que se adota. Ou seja, quanto maior o vocabulário mais difícil para o sistema reconhecer sentenças [16] [17] [15].

3.5 Variabilidades do Sinal de Voz

O reconhecimento de fala é um problema difícil devido às várias fontes de variabilidades associadas ao sinal de voz:

· Variabilidades fonéticas: Os fones são bastante sensíveis ao contexto pois podem modificar o sentido das sentenças. A exemplo da frase 'nada mais é...' que pode ser pronunciada e compreendida como 'nada mais zé'. Em vocabulários extensos como o português, é comum ter que lidar com palavras com a mesma pronúncia (homófonas) [16] [17].

· Variabilidades acústicas: Podem resultar de mudanças no ambiente assim como da posição e características do transdutor(microfone). Essas características são difíceis de contornar principalmente em ambientes abertos onde o ruído é inevitável [16] [17].

· Variabilidades intra-locutor: Podem resultar de mudanças do estado físico/emocional dos locutores, ritmo, timbre e intensidade da voz do locutor [16] [18] [17].

· Variabilidades entre-locutores: Originam-se da variação linguística da língua nas diversas regiões do país, assim como do tamanho e forma do trato vocal [16] [3] [17].

3.6 Estado da Arte

Atualmente, os reconhecedores de fala contínua com amplo vocabulário utilizam modelos híbridos no desenvolvimento do modelo acústico e de linguagem. Independente do tipo de estrutura utilizada, a literatura apresenta a seguinte arquitetura na estruturação

de reconhecedores de fala: interface de captação e extração de parâmetros do sinal de fala, modelo acústico, modelo linguístico e decodificador [19] [20] [9] [16] [3] [21] [17].

É possível observar que o uso do HMM (*Hidden Markov Models* - Modelos Ocultos de Markov) ainda fornece uma vasta aplicação em tarefas de reconhecimento de padrões devido aos seus bons resultados e seu uso na modelagem do sinal de voz. Contudo, modelos híbridos que combinam o HMM com o DNN (*Deep Neural Network*) têm apresentado um importante papel no desenvolvimento dos reconhecedores de fala [21] [17] [19].

3.7 Aplicações do Reconhecedor

Visto que aplicações do reconhecedor de fala proporcionam uma maior produtividade por auxiliarem ou substituem operadores humanos, pode-se listar algumas áreas para sistemas de reconhecimento automático da fala, tais como: ditado, sistemas de interface e transcritor, serviços de telefonia automáticos, aplicações industriais, discriminação de vozes patológicas e identificação de emoções [19] [16] [3] [15].

3.7.1 Desafios do Reconhecimento de Fala

Levando em consideração esses fatos, as dificuldades em se trabalhar com sinais de fala, são: [10]

- Dificuldade em se remover ruídos.
- Definir com precisão o início e o fim de uma palavra.
- Palavras que possuem o mesmo som, porém com significados diferentes(homônimos).
- Estrutura gramatical e semântica das palavras.

3.8 Modelo Oculto de Markov

O HMM (*Hidden Markov Model* - Modelo Oculto de Markov) é um modelo estatístico baseado na teoria dos processos de Andrei Markov, utilizado para modelar processos estocásticos. Essencialmente, o HMM é uma máquina de estados finita que modela as transições entre os estados e que usa um conjunto de funções densidade de probabilidade (fdp) parra modelar as variações aleatórias do sinal em cada estado. Nesse processo de dupla camada, existe uma camada oculta de Markov controlando o estado de uma camada observável [21] [16] [22] [8].

É comum a aplicação de HMMs em sistemas de reconhecimento de fala para modelar palavras e até unidades menores como os fones [23].

Os Modelos Ocultos de Markov apresentam um algoritmo eficiente e robusto para as tarefas de treinamento e reconhecimento. O treinamento é responsável por modelar o conjunto de parâmetros acústicos extraídos do sinal de voz, pertencentes às observações, por uma sequência de estados de acordo com a variação temporal do sinal de voz [23].

Dessa maneira, caracteriza-se um HMM por um conjunto de N estados que estão conectados por transições. A cada instante (t) há uma mudança de estado com uma determinada densidade de probabilidade em sua saída . A sequência de símbolos emitidos

(observações) representam a saída do HMM. Durante o processamento do HMM, o estado ou valor a qualquer momento (t), depende de seu estado anterior e valores no tempo $(t-1)$, e é independente do histórico do processo antes de $(t-1)$ [22] [8] [23].

3.8.1 Parâmetros do HMM

Os parâmetros associados para especificação do HMM são: N, K, A, B e Π. Assim definidos [21] [3] [8] [11]:

N : Representa o número de estados do modelo.

K : Representa o número de símbolos no alfabeto, quando o espaço é definido por uma fdp discreta ou número de grupos quando for uma fdp contínua.

$\mathbf{A} = a_{ij}$: Matriz de probabilidades de transições do estado i para o estado j; sendo $a(i,j) = P(Z_t = j/Z_{t-1} = i)$, $1 \leq i, j \leq N$.

$\mathbf{B} = b_{jk}$: Representa a matriz de distribuição da probabilidade de observação em cada estado, ou seja, a probabilidade de emissão do símbolo 'k' no estado 'j', assim, $b(j,k) = P(X_t = k/Z_t = j)$ em que $X = x_1, x_2, ..., x_T$ é a sequência observada.

$\Pi = \Pi_i$: É o vetor de distribuição de estado inicial do modelo começar no estado i, assim, $\Pi_i = P(Z_1 = i)$, $1 \leq i \leq N$.

No entanto, é possível formalizar uma forma compacta que indica todo o conjunto de parâmetros necessários para o modelo, definido por [23] [21] [12] [11] [8]:

$$\lambda = (\Pi, A, B). \tag{1}$$

3.8.2 Classificação do Modelo Oculto de Markov

Os Modelos Ocultos de Markov podem ser classificados tanto por sua distribuição de probabilidade (contínua ou discreta) quanto por sua topologia [23] [21] [8].

3.8.3 Caso Discreto

A cada estado do HMM, é associada uma função massa de probabilidade. Durante esse processo, as sequências de observações são formadas por índices de vetores de um dicionário. Dessa maneira, há um número finito de símbolos de saída K cujas propriedades consistem em [12] [8] [13]:

$$B(j,k) \geq 0, \ para \ 1 \leq j \leq N \ e \ 1 \leq k \leq K, \tag{2}$$

$$\sum_{k=1}^{K} B(j,k) = 1, \tag{3}$$

em que:

- N é o número de estados do HMM;
- K é o número de símbolos discretos do modelo;
- B(j,k) é a probabilidade de emissão do símbolo k no estado j.

3.8.4 Caso Contínuo

Para o caso contínuo associa-se a cada estado do HMM a uma função densidade de probabilidade. É comum usar uma função densidade de probabilidade modelada por uma mistura de K gaussianas multidimensionais [12] [19] [8] :

$$B(j,t) = \sum_{k=1}^{K} R(j,k) \cdot G((x(t), \mu(j,k), \Sigma(j,k)), \tag{4}$$

em que:

- x(t) é o vetor de parâmetros de entrada de dimensão D no instante de tempo t;
- K é o número de gaussianas da mistura para cada estado;
- R(j,k) é o coeficiente de ponderação da k-ésima mistura no estado j;
- G é a função densidade de probabilidade gaussiana multidimensional com vetor média $\mu(j,k)$ e matriz de covariância $\Sigma(j,k)$ para o componente da k-ésima mistura no estado j.

A aproximação mais comum para o cálculo da densidade de probabilidade das observações é a partir de uma distribuição Gaussiana que modele o espaço acústico das observações no estado. Uma forma bastante popular dessa fdp é a utilização de uma mistura de Gaussianas. A mistura é uma soma linearmente ponderada de diferentes densidades Gaussianas que representam a distribuição de probabilidade das observações no estado. A função densidade de probabilidade gaussiana multidimensional é dada por [21] [24] [18] [12] [19]:

$$G((x(t), \mu(j,k), \Sigma(j,k)) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} e^{-\frac{1}{2}(x(t)-\mu(j,k))' \Sigma^{-1} (x(t)-\mu(j,k))}, \tag{5}$$

em que:

- D é a dimensão do vetor x(t);
- $|\Sigma|$ é o determinante da matriz de covariância;
- Σ^{-1} é a inversa da matriz de covariância.

3.8.5 Topologia dos HMMs

A topologia de um HMM é caracterizada pelas transições de estados de acordo com a matriz de transição **A**. São as probabilidades de transição dessa matriz que definem o processo de Markov e sua ordem [23] [21] [25].

Dessa maneira, pode-se classificar as topologias como ergódicas (totalmente conectadas) ou *left-right* (modelo de Bakis) [23] [21] [25] [11].

Um HMM do tipo ergódico não é apropriado para uma boa representação da voz visto que essa topologia não modela bem as características temporais dos eventos acústicos em cada estado; o que gera o risco de convergência em máximos locais [23] [21] [25].

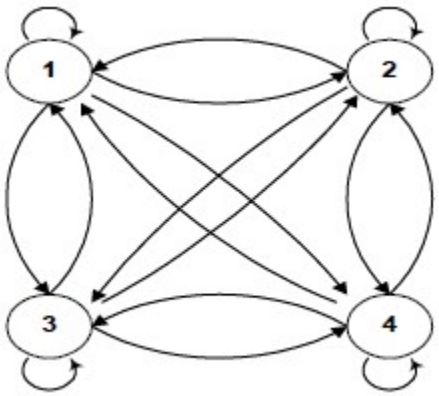

Figura 4: Modelo ergódico de 4 estados [21].

Em um modelo ergódico, não há restrição de transição de estados. Sua matriz de transição é:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

No modelo *left-right* é possível observar, como principal característica, o efeito sequencial da transição de estados, ou seja, dado que a cadeia de Markov deixa um estado no tempo (t), o mesmo não pode ser visitado novamente a partir de $(t+1)$ [23] [21] [11].

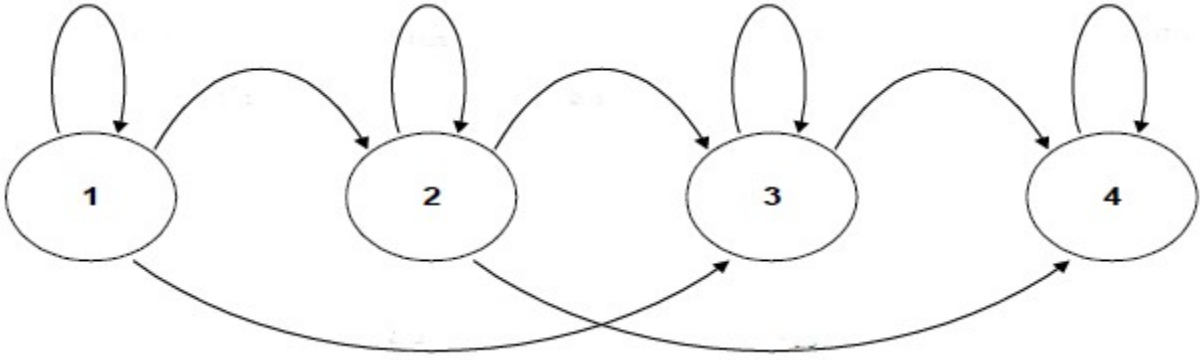

Figura 5: Modelo *left-right* de 4 estados [21].

Matricialmente, o modelo *left-right* para N = 4 estados é:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix}$$

Portanto, dada a característica sequencial da fala e dos melhores resultados apresentados quando comparados aos modelos ergódicos, o modelo *left-right* é a melhor escolha para a tarefa de reconhecimento de fala [11] [21].

3.9 Os 3 problemas de um HMM

A modelagem de um HMM segue a resolução de 3 problemas [23] [24] [13] [26]:

- Problema de Avaliação : Encontrar a probabilidade de uma sequência, ou seja, dado um modelo λ e uma sequência de observações, busca-se qual a probabilidade dessas observações terem sido geradas por esse modelo [23] [13] [24] [26].
- Problema de Decodificação : Por meio de um modelo e uma sequência de observações, busca-se encontrar qual a sequência de estados que explica essas observações [23] [24] [13] [26].
- Problema de Treinamento : Dado um modelo e uma sequência de observações (ou um conjunto de sequências de observações), busca-se ajustar os parâmetros do modelo λ = (Π, A, B) [23] [24] [13] [26].

3.9.1 Solução do Problema de Avaliação

O objetivo é encontrar a probabilidade de uma sequência de observações $X = (x_1, x_2, ..., x_T)$, ou seja, encontrar a $P(X)$. O método mais imediato de que dispõe-se, é somar todas as probabilidades de sequência de estado possíveis [13] [27]:

$$P(X) = P(x_1, x_2, ..., x_T). \tag{6}$$

Expande-se para obter a distribuição de probabilidade conjunta:

$$P(X) = \sum_{z_1=1}^{N} \sum_{z_2=1}^{N} ... \sum_{z_T=1}^{N} P(x_1, x_2, ..., x_T, z_1, z_2, z_T). \tag{7}$$

Aplica-se a regra de Bayes para fatorar essa distribuição de probabilidade conjunta em distribuições condicionais:

$$P(X) = \sum_{z_1=1}^{N} \sum_{z_2=1}^{N} ... \sum_{z_T=1}^{N} P(z_1)P(x_1/z_1) \prod_{t=2}^{T} P(z_t/z_{t-1})P(x_t/z_t). \tag{8}$$

Reescrevendo essas probabilidades para os parâmetros de acordo com o modelo λ:

$$P(X) = \sum_{z_1=1}^{N} \sum_{z_2=1}^{N} ... \sum_{z_T=1}^{N} \Pi(z_1)B(x_1/z_1) \prod_{t=2}^{T} A(z_t/z_{t-1})B(x_t/z_t). \tag{9}$$

Apesar de simples, essa operação demanda um grande esforço computacional de crescimento exponencial: $O(TN^T)$. O algoritmo *forward* é uma solução de programação dinâmica mais eficiente. Ele armazena até N valores a cada etapa do tempo e reduz a complexidade computacional para $O(N^2T)$ [13] [26].

O algoritmo *forward*

Considere T=3 para a equação 8. Separa-se os somatórios e seus termos e obtém-se:

$$P(X) = \sum_{z_3=1}^{N} P(x_3/z_3) \sum_{z_2=1}^{N} P(x_2/z_2)P(z_3/z_2) \sum_{z_1=1}^{N} P(x_1/z_1)P(z_2/z_1)P(z_1). \tag{10}$$

Agrupa-se os termos de forma recursiva, exceto no tempo t=T:

$$P(x_{t+1}/z_{t+1}) \sum_{z_t=1}^{N} P(z_{t+1}/z_t)(P(x_t/z_t) \sum_{z_{t-1}=1}^{N} ...). \tag{11}$$

Faça:

$$\alpha(t, z_t) = (P(x_t/z_t) \sum_{z_{t-1}=1}^{N} ...). \tag{12}$$

Então:

$$\alpha(t+1, z_{t+1}) = P(x_{t+1}/z_{t+1}) \sum_{z_t=1}^{N} P(z_{t+1}/z_t)\alpha(z_t). \tag{13}$$

A representação desses passos em termos dos parâmetros do HMM é [13] [26] [27] [28]:

- Passo de Inicialização

$$\alpha(t, i) = \Pi_i B(i, x_t), \ para\ t = 1. \tag{14}$$

- Passo de Indução

$$\alpha(t+1, j) = B(j, x_{t+1}) \sum_{i=1}^{N} \alpha(t, i)A(i, j), \ para\ t = 1, ..., T-1. \tag{15}$$

- Passo de Terminação

$$P(X) = \sum_{i=1}^{N} \alpha(T, i). \tag{16}$$

Note que a variável α representa a probabilidade conjunta de x_1 a x_t no estado oculto i [26].

Com esses 3 passos realizados, tem-se o algoritmo *forward* e soluciona-se o primeiro problema do HMM [27] [26] [28].

O algoritmo *backward*

Apesar de não ser necessário no problema de avaliação, o uso do algoritmo *backward* é justificado no problema de treinamento. Entretanto, é importante observar a relação entre os algoritmos *forward-backward* [28].

Define-se a variável β como a probabilidade de ver a sequência de observações nos próximos passos de tempo a partir do estado atual. Dessa maneira, define-se os seguintes passos [13] [26] [28]:

- Passo de Inicialização

$$\beta(T, i) = 1. \tag{17}$$

  Dado que não há observações futuras após o último passo T, adota-se o valor de probabilidade como 1 [13].

- Passo de Indução

$$\beta(t, i) = \sum_{j=1}^{N} A(i, j)B(j, x_{t+1})\beta(t+1, j), \ para\ t = T-1, ..., 1. \tag{18}$$

Note que para a variável β não há a necessidade do passo de Terminação.

Dessa forma, a variável α considera a probabilidade conjunta de x_1 a x_t enquanto que β considera a probabilidade conjunta de x_{t+1} a x_T. Em outras palavras, pode-se afirmar que α considera as observações do início à observação atual, enquanto que β considera a partir da observação imediatamente seguinte até a última [21].

3.9.2 Solução do Problema de Decodificação

O objetivo é encontrar a sequência mais provável de estados ocultos Z dada uma sequência de observações X, ou seja, pretende-se calcular o melhor caminho. A principal técnica de programação dinâmica para resolver esse problema é a partir do algoritmo de Viterbi. Considere $\delta(t, j)$ como a máxima probabilidade conjunta de chegar no estado 'j' no tempo 't', considerando todos os estados anteriores. Verifica-se os seguintes passos [13] [26] [27] [28]:

- Passo de Inicialização

$$\delta(1, j) = \Pi_j B(j, x(1)). \tag{19}$$

- Passo de Indução

$$\delta(t, j) = \max_{i \in 1...N} \delta(t-1, i) A(i, j) B(j, x(t)). \tag{20}$$

- Passo de Terminação

$$P^* = \max_{j \in 1...N} \delta(T, j). \tag{21}$$

Note que esse algoritmo é similar ao *forward* desenvolvido. A variável $\delta(T, j)$ representa a probabilidade de estar no estado 'j' no tempo final 'T'. Dessa maneira, o passo de terminação calcula a probabilidade para a melhor sequência de estados ocultos. Entretanto, também é necessário descobrir quais foram esses estados ocultos [13] [26] [27].

Adiciona-se a variável $\psi(t, j)$ que define no tempo 't' qual é o melhor estado anterior que leva ao estado 'j'. Dessa forma, desenvolve-se os seguintes passos [13] [27] [28]:

- Passo de Inicialização
  Dado que no tempo inicial 't=1' ainda não existe estados anteriores, tem-se:

$$\psi(1, j) = 0. \tag{22}$$

- Passo de Indução
  A partir do passo de indução:

$$\delta(t, j) = \max_{i \in 1...N} \delta(t-1, i) A(i, j) B(j, x(t)). \tag{23}$$

  Desenvolve-se:

$$\psi(t, j) = \max_{i \in 1...N} \delta(t-1, i) A(i, j). \tag{24}$$

Note que não precisa da matriz $\mathbf{B}$ visto que a mesma é independente do estado anterior [27].

- Passo de Terminação

  A partir de:

$$P^* = \max_{j \in 1...N} \delta(T, j). \tag{25}$$

  Obtém-se:

$$Z(T)^* = \underset{i \in 1...N}{\text{argmax}}\ \delta(T, i). \tag{26}$$

  Retorna o melhor estado no final da sequência

- Recuo

  Após executar todo o algoritmo desenvolvido, é preciso recuperar a sequência de estado ideal. Repete-se esses passos, para $t = T, ..., 1$ até chegar no estado inicial [27] [28]:

$$Z(t)^* = \psi(t + 1, Z(t + 1)^*). \tag{27}$$

3.9.3 Solução do Problema de Treinamento

Nesse caso, pretende-se estimar os parâmetros Π, A, B de um modelo λ que melhor descreva a sequência de observações. Na solução desse problema, é esperado que o método maximize as probabilidades dos dados de treinamento. Para isso, utiliza-se o algoritmo iterativo *forward-backward* também conhecido como algoritmo de Baum-Welch [28] [13].

Intuitivamente, pode-se computar os parâmetros do HMM da seguinte maneira [28] [13]:

- $\Pi(i)$ é a contagem de quantas vezes o primeiro estado foi 'i' sobre o número total de sequências de observações X.

$$\Pi(i) = \frac{contador(Z_1 = i)}{M}. \tag{28}$$

- $A(i, j)$ é a contagem de quantas vezes transitou-se do estado 'i' para o estado 'j', sobre a quantidade de vezes que esteve-se no estado 'i'.

$$A(i, j) = \frac{contador(Z_t = i \rightarrow Z_{t+1} = j)}{contador(Z_t = i)}. \tag{29}$$

- $B(j, k)$ é a contagem de quantas vezes esteve-se no estado 'j' e emitiu-se o símbolo 'k', sobre a quantidade de vezes que esteve-se no estado 'j'.

$$B_{j,k} = \frac{contador(Z_t = j \wedge x_t = k)}{contador(Z_t = j)}. \tag{30}$$

Reescrevendo os contadores definidos em 28, 29 e 30:

$$\Pi(i) = \frac{contador(Z_1 = i)}{M} = P(Z_1 = i), \tag{31}$$

$$A(i,j) = \frac{contador(Z_t = i \rightarrow Z_{t+1} = j)}{contador(Z_t = i)} = \frac{contador(Z_t = i \rightarrow Z_{t+1} = j)/M}{contador(Z_t = i)/M}, \tag{32}$$

$$A(i,j) = \frac{P(Z_t = i, Z_{t+1} = j)}{P(Z_t = i)}, \tag{33}$$

$$B(j,k) = \frac{contador(Z_t = j \wedge x_t = k)}{contador(Z_t = j)} = \frac{contador(Z_t = j \wedge x_t = k)/M}{contador(Z_t = j)/M}, \tag{34}$$

$$B(j,k) = \frac{P(Z_t = j, x_t = k)}{P(Zt = j)}. \tag{35}$$

Nota-se que em $A(i,j)$ e $B(j,k)$, o numerador é uma probabilidade conjunta, enquanto que o denominador é uma probabilidade marginal. Dessa maneira, $P(Z_t = i, Z_{t+1} = j)$ representa o número de vezes que transitou-se do estado 'i' para o estado 'j' no tempo 't' e, por consequência, $\sum_{t=1}^{T-1} P(Z_t = i, Z_{t+1} = j)$ representa o número total de vezes que ocorreu a mesma transição em questão [13] [28].

Por isso, reescreve-se os parâmetros como [28] [13]:

$$\Pi(i) = P(Z_1 = i), \tag{36}$$

$$A(i,j) = \frac{\sum_{t=1}^{T-1} P(Z_t = i, Z_{t+1} = j)}{\sum_{t=1}^{T-1} P(Z_t = i)}, \tag{37}$$

$$B(j,k) = \frac{\sum_{t=1}^{T} P(Z_t = j, x_t = k)}{\sum_{t=1}^{T} P(Z_t = j)}. \tag{38}$$

Percebe-se que como a matriz A_{ij} considera as transições, o somatório deve compreender o intervalo entre $t = 1, ..., T-1$ visto que a última transição ocorre no penúltimo termo. A matriz B_{jk} não depende de transições, mas do estado 'j' em que se emite o símbolo 'k'; logo compreende todo o tempo $t = 1, ..., T$ [21].

Define-se então, duas novas variáveis: $\xi_t(i,j)$ e $\gamma_t(i)$ [28] [13].

A primeira é a probabilidade de estar no estado i no tempo t e no estado j no tempo $t+1$, definida por [13] [28]:

$$\xi_t(i,j) = P(Z_t = i, Z_{t+1} = j/X) = \frac{P(Z_t = i, Z_{t+1} = j, X)}{P(X)} = \frac{P(Z_t = i, Z_{t+1} = j, X)}{\sum_{i=1}^{N}\sum_{j=1}^{N} P(Z_t = i, Z_{t+1} = j, X)}. \tag{39}$$

Desenvolvendo a equação 39 em termos conhecidos pelo algoritmo de Baum-Welch, tem-se [28] [13]:

$$\xi_t(i,j) = \frac{\alpha_t(i)A(i,j)B(j,x(t+1))\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)A(i,j)B(j,x(t+1))\beta_{t+1}(j)}. \tag{40}$$

A segunda variável diz respeito a probabilidade de estar no estado i dada a sequência de observação X [13] [28].

$$\gamma_t(i) = P(Z_t = i/X) = \sum_{j=1}^{N}\xi_t(i,j). \tag{41}$$

Reestimação de Parâmetros

A partir das variáveis $\xi_t(i,j)$ e $\gamma_t(i)$ definidas, pode-se formalizar a reestimação dos parâmetros do modelo λ [27]:

$$\Pi(i) = \gamma_1(i) = P(Z_1 = i/X), \tag{42}$$

$$A(i,j) = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, \tag{43}$$

$$B(j,k) = \frac{\sum_{t=1}^{T}P(Z_t = j, x_t = k/X)}{\sum_{t=1}^{T}P(Z_t = j/X)} = \frac{\sum_{t=1}^{T}\gamma_t(j)1(x_t = k)}{\sum_{t=1}^{T}\gamma_t(j)}. \tag{44}$$

Observe que $1(x_t = k)$ é a função indicadora que retorna 1 caso $x_t = k$ ou 0 caso contrário.

Neste ponto, tem-se dois grupos de cálculo: ξ, γ e Π, A, B. Dessa forma, adquire-se uma dependência circular [29] [27] .

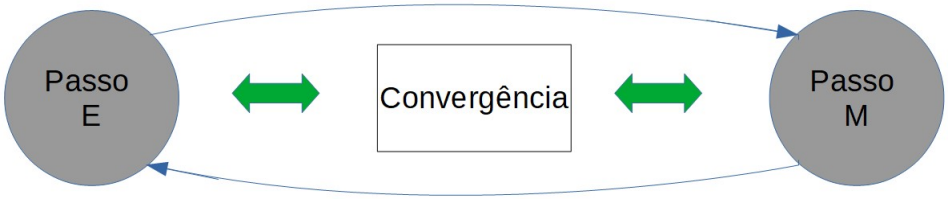

Figura 6: Passos E e M [Autoria própria].

Para resolver isso, utiliza-se o algoritmo de maximização de expectativas (Expectation - Maximization) para o HMM que é o algoritmo de Baum-Welch. Dado que esse procedimento leva a um máximo local, é necessário realizar os cálculos do passo E e passo M até conseguir uma convergência [29] [27].

O passo E consiste em calcular as estimativas iniciais ξ e γ e o passo M consiste em reestimar os parâmetros para calcular Π, A e B. Esse processo repete-se em *loop* até um ótimo local [29] [27].

Graficamente, o passo E fornece o ponto inicial ao qual fornece os parâmetros necessários para encontrar uma função de limite inferior que toca a verdadeira função objetivo nessa configuração de parâmetros. O passo M é onde maximiza-se a função de limite inferior em relação aos parâmetros [29] [27].

3.9.4 Atualizações de Baum-Welch para Múltiplas Observações

Até este ponto, considerou-se apenas as estimativas para uma única sequência de observação. Considere, portanto, múltiplas observações para $m = 1..M$ em que cada observação m tem tamanho $T(m)$. Dessa maneira, cada observação é indexada por $'m'$. Reescreve-se as atualizações em termos de α e β [21] [29]:

$$\Pi(i) = \frac{1}{M}\sum_{m=1}^{M}\frac{\alpha_m(1,i)\beta_m(1,i)}{P(m)}, \tag{45}$$

$$A(i,j) = \frac{\sum_{m=1}^{M}\frac{1}{P(m)}\sum_{t=1}^{T(m)-1}\alpha_m(t,i)A(i,j)B(j,x_m(t+1))\beta_m(t+1,j)}{\sum_{m=1}^{M}\frac{1}{P(m)}\sum_{t=1}^{T(m)-1}\alpha_m(t,i)\beta_m(t,i)}, \tag{46}$$

$$B(j,k) = \frac{\sum_{m=1}^{M}\frac{1}{P(m)}\sum_{t=1}^{T(m)}\alpha_m(t,j)\beta_m(t,j),\ \text{se } x_m(t)=k, \text{ caso contrário é } 0}{\sum_{m=1}^{M}\frac{1}{P(m)}\sum_{t=1}^{T(m)}\alpha_m(t,j)\beta_m(t,j)}. \tag{47}$$

3.9.5 O Problema do *Underflow*

O *underflow* surge quando se trabalha com números muito pequenos a ponto do computador computá-los como zero. Dado que o resultado das probabilidades trabalhadas resultam em números muito pequenos, há o risco de ocorrer esse problema. Dessa forma, é preciso adaptar um fator de dimensionamento para os cálculos do algoritmo de Baum-Welch. Dado um fator de dimensionamento $c(t)$ em que $t = 1, ..., T$, tem-se [27]:

- Algoritmo *Forward*
  - Inicialização

$$\alpha'(1,i) = \Pi(i)B(j,x(1)) = \alpha(1,i), \tag{48}$$

$$c(t) = \sum_{i=1}^{N}\alpha'(t,i), \tag{49}$$

$$\hat{\alpha}(t,i) = \frac{\alpha'(t,i)}{c(t)}, \tag{50}$$

$$\hat{\alpha}(1,i) = \frac{\Pi(i)B(j,x(1))}{c(1)}. \tag{51}$$

– Indução

$$\alpha'(t,j) = \sum_{i=1}^{N} \hat{\alpha}(t-1,i)A(i,j)B(j,x(t)), \tag{52}$$

$$\hat{\alpha}(t,i) = \frac{\alpha'(t,i)}{c(t)}. \tag{53}$$

– Terminação

$$P(X) = \prod_{t=1}^{T} c(t). \tag{54}$$

- Algoritmo *Backward*

– Inicialização

$$\hat{\beta}(T,i) = 1. \tag{55}$$

– Indução

$$\hat{\beta}(t,i) = \frac{\sum_{j=1}^{M} A(i,j)B(j,x(t+1))\hat{\beta}(t+1,j)}{c(t+1)}. \tag{56}$$

As novas atualizações para múltiplas observações e considerando um fator de dimensionamento para resolver o *underflow* são [29]:

$$\Pi(i) = \frac{1}{M}\sum_{m=1}^{M} \hat{\alpha}_m(1,i)\hat{\beta}_m(1,i), \tag{57}$$

$$A(i,j) = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T(m)-1} \frac{\hat{\alpha}_m(t,i)\hat{\beta}_m(t+1,j)}{c(t+1)} A(i,j)B(j,x_m(t+1))}{\sum_{m=1}^{M}\sum_{t=1}^{T(m)-1} \hat{\alpha}_m(t,i)\hat{\beta}_m(t,i)}, \tag{58}$$

$$B(j,k) = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T(m)} \hat{\alpha}_m(t,j)\hat{\beta}_m(t,j), \text{ se } x_m(t)=k, \text{ caso contrário é } 0}{\sum_{m=1}^{M}\sum_{t=1}^{T(m)} \hat{\alpha}_m(t,j)\hat{\beta}_m(t,j)}. \tag{59}$$

3.9.6 Modelo de Mistura de Gaussianas

O algoritmo de Baum-Welch, descrito até o momento, utilizou do caso discreto como demonstração de sua construção. No entanto, para o reconhecimento de fala, é importante utilizar o caso contínuo com o uso de GMM (*Gaussian Mixture Model* - Modelo de Mistura de Gaussianas). Assim como observado na equação 4, o GMM faz a substituição da matriz de probabilidade de emissões B por três novas variáveis: R, μ e Σ. Dessa forma, a equação 4 armazena componentes individuais da mistura por [27] [13] [21] [29]:

$$Comp(j,k,t) = R(j,k)G(x(t),\mu(j,k),\Sigma(j,k)). \tag{60}$$

Os passos do algoritmo EM são dados por [29] [13] [27]:

- Passo E

$$\gamma(j,k,t) = \frac{\alpha(t,j)\beta(t,j)}{\sum_{j'=1}^{N} \alpha(t,j')\beta(t,j')} \frac{R(j,k)G(x(t),\mu(j,k),\Sigma(j,k))}{\sum_{k'=1}^{K} R(j,k')G(x(t),\mu(j,k'),\Sigma(j,k'))}, \tag{61}$$

$$\gamma(j,k,t) = \frac{\alpha(t,j)\beta(t,j)}{\sum_{j'=1}^{N} \alpha(t,j')\beta(t,j')} \frac{Comp(j,k,t)}{B(j,t)}. \tag{62}$$

- Passo M

$$R(j,k) = \frac{\sum_{t=1}^{T} \gamma(j,k,t)}{\sum_{t=1}^{T}\sum_{k'=1}^{K} \gamma(j,k',t)}, \tag{63}$$

$$\mu(j,k) = \frac{\sum_{t=1}^{T} \gamma(j,k,t)x(t)}{\sum_{t=1}^{T} \gamma(j,k,t)}, \tag{64}$$

$$\Sigma(j,k) = \frac{\sum_{t=1}^{T} \gamma(j,k,t)(x(t)-\mu(j,k))(x(t)-\mu(j,k))^T}{\sum_{t=1}^{T} \gamma(j,k,t)}. \tag{65}$$

3.10 Modelo de Linguagem

O modelo de linguagem é um componente do sistema de reconhecimento de fala que estima as probabilidade $P(W)$ de possíveis enunciados falados. Essas probabilidades são combinadas com as probabilidades do modelo acústico $P(X/W)$ na equação fundamental do reconhecimento de fala para chegar à melhor hipótese geral. Isso pode ser verificado pela equação 66 [30] [16].

$$\hat{W} = \underset{w}{\text{argmax}}[P(X/W)P(W)]. \tag{66}$$

Desse modo, o modelo de linguagem incorpora-se ao reconhecedor sobre quais são as prováveis sequências de palavras. Em vez de codificar regras rígidas de sintaxe e semântica da língua, deve-se atribuir altas probabilidades às prováveis sequências e baixas probabilidades às improváveis, sem descartar completamente nenhuma [16] [19] [31] [17].

É importante observar que, no modelo de linguagem, frequentemente usa-se sentenças como a sequência de palavras correspondente a toda uma locução, sem sugerir que a mesma represente algo como uma sentença correta e completa de acordo com as regras gramaticais convencionais [16] [31] [9].

3.10.1 Vocabulário

Considere W como uma sentença, ou seja, um sequência de palavras dadas por: $W = w_1 w_2 ... w_n$. Em que 'n' é o número de palavras que é ilimitado a princípio. Na simplificação do problema de limitar a escolha de palavras a um conjunto finito, é que define-se um vocabulário [16] [19].

Palavras fora do vocabulário podem gerar erros de reconhecimento. Por isso, é importante a escolha do vocabulário para a minimização de erros. Uma estratégia é escolher as palavras com mais frequência em um conjunto de dados de treinamento. Dessa forma, é preciso considerar o tamanho do vocabulário pois o mesmo afeta sua computação na decodificação e sua precisão [16] [19].

3.10.2 Modelo de Markov e N-grama

Mesmo com um vocabulário finito, ainda existe um conjunto infinito de sequências de palavras; portanto, não se pode parametrizar o modelo de linguagem listando a probabilidade de cada sentença possível. Apesar de, teoricamente, ser possível fazer isso, seria muito difícil obter estimativas de probabilidade confiáveis, pois a maioria das frases possíveis seriam muito raras [30] [31] [16].

Para solucionar esse problema, usa-se a regra da cadeia para fatorar a probabilidade de sentença em um produto de probabilidades condicionais de palavra e, em seguida, aplicar a suposição de Markov para limitar o número de estados. Supondo o uso de um modelo de Markov limitado ao contexto de uma palavra, tem-se [9] [19] [21]:

$$P(W) = P(w_1)P(w_2/w1)P(w_3/w_2)...P(w_n/w_{n-1}). \tag{67}$$

Nesse contexto, usa-se um modelo de Markov de primeira ordem. No entanto, no modelo de linguagem, essa terminologia não é usada e, em vez disso, chama-se de modelo bigrama porquê usa-se apenas estatísticas de duas palavras adjacentes por vez. Do mesmo modo, um modelo de Markov de segunda ordem é chamado de modelo trigrama e preveria cada palavra com base nas duas anteriores conforme a equação 68 [31] [9].

$$P(W) = P(w_1)P(w_2/w_1)P(w_3/w_2, w_1)...P(w_n/w_{n-1}, w_{n-2}). \tag{68}$$

A generalização desse esquema é o modelo N-grama, ou seja, cada palavra é condicionada às $N-1$ palavras anteriores. Segundo a literatura, há uma boa melhoria ao passar de bigramas para trigramas mas pouca melhoria à medida que N é aumentado ainda mais. Portanto, na prática, raramente usa-se modelos acima do trigrama [31] [16] [19] [21].

4 Objetivos

Objetivo Geral:

Aprimoramento de um Sistema de Reconhecimento de Fala para ser aplicado em um conversor português-LIBRAS. Implementação de um sistema que realize a conversão dos idiomas português-LIBRAS utilizando processamento de fala para facilitar a comunicação entre pessoas com e sem algum grau de surdez.

Objetivos Específicos:

- Estudar a sintaxe e semântica da LIBRAS;

- Implementar um sistema de reconhecimento de fala;

- Aprimorar o algoritmo de mapeamento entre texto e imagem;

- Estudar e implementar sinais em LIBRAS que representem ações;

- Aprimorar os conhecimentos na linguagem de programação Python;

- Realizar a conversão entre a fala reconhecida e imagens do idioma LIBRAS;

- Testes de análise de desempenho.

5 Metodologia

5.1 Sistema Proposto

O diagrama de blocos da Figura 7 ilustra a construção do sistema proposto.

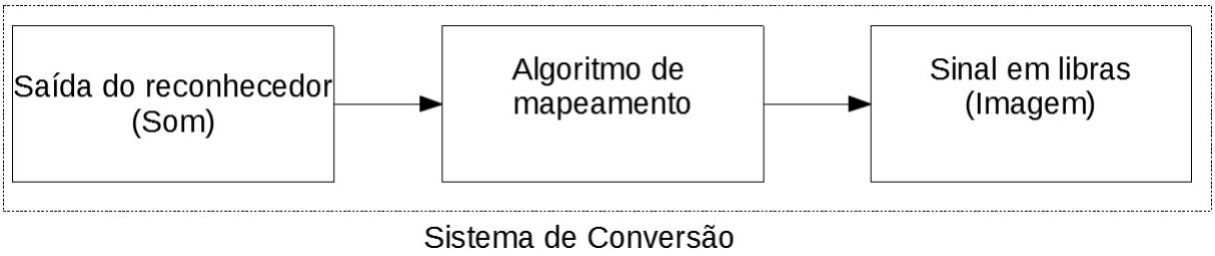

Figura 7: Diagrama de blocos do sistema de conversão [Autoria própria].

Após a aquisição do sinal gerado pelo reconhecedor de fala, mapeia-se o sinal para gerar as imagens que correspondem àquela palavra ou letra. Esse processo final utiliza de um algoritmo de busca para vincular letra/imagem ou palavra/imagem. Neste trabalho, utilizou-se a linguagem de programação *Python* por ser uma linguagem livre e multiplataforma.

Dessa maneira, a maior parte do trabalho se encontra na produção de um bom reconhecedor de fala que possa gerar informações confiáveis para o conversor gerar o real sinal em LIBRAS.

5.2 Etapas do Reconhecimento de Fala

Pode-se definir um HMM como uma sequência de estados S_n e observações O_t. No treinamento, é realizada a associação de um estado com suas probabilidades de transições entre os estados e as probabilidades de observações dado o estado. A sequência temporal

é segmentada pelos estados obedecendo a algum evento acústico que será função da estrutura do modelo utilizado dado que as observações são os vetores de características da voz abordados. A Figura 8, ilustra esse processo [17] [21] [23] [16].

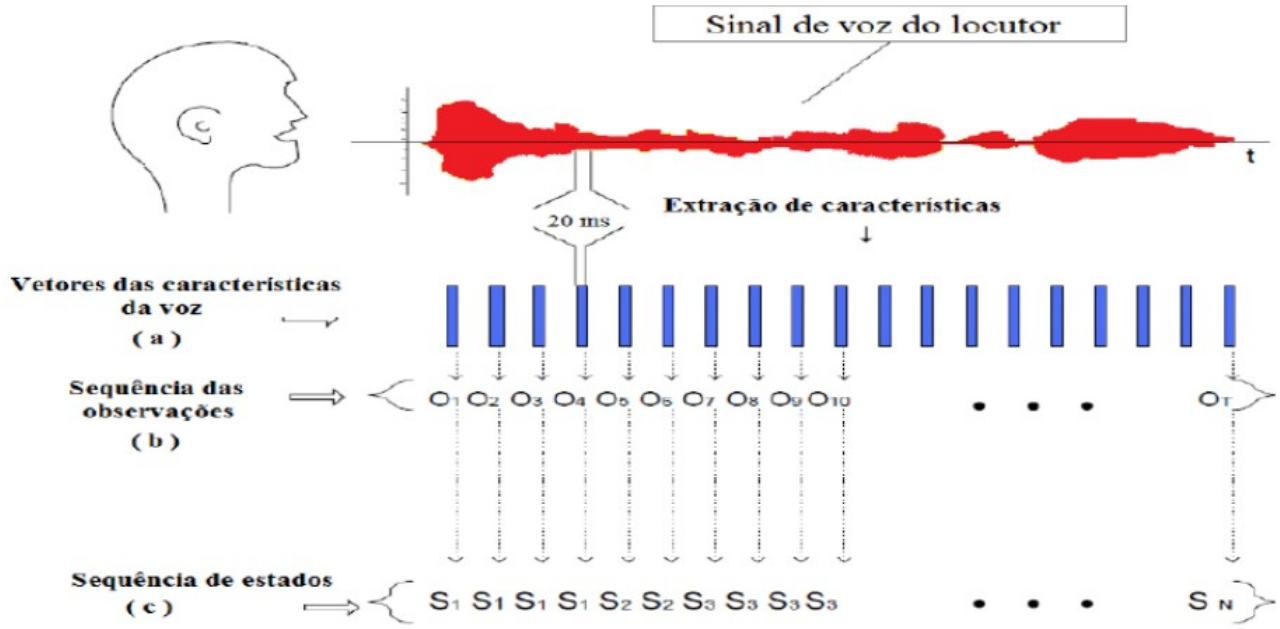

Figura 8: (a) Vetores das características da voz do locutor; (b) Sequência das observações; (c) Sequência de estados [21].

Portanto, o sistema de reconhecimento de fala tem por objetivo identificar qual a palavra ou frase foi pronunciada pelo locutor. Dessa forma, o reconhecimento de padrões precisa de uma fase de treinamento e outra de reconhecimento a fim de, inicialmente, gerar uma base de dados de treinamento (fonemas, trifones, palavras, frases) como modelos de referência para o que pretende ser reconhecido. Na etapa de reconhecimento, os modelos obtidos da fase de treinamento são usados para comparação cuja regra de decisão estipula o que mais se assemelha àquele padrão [32] [33] [23].

A partir da Figura 9 , é possível observar todas as etapas necessárias para o reconhecimento de fala.

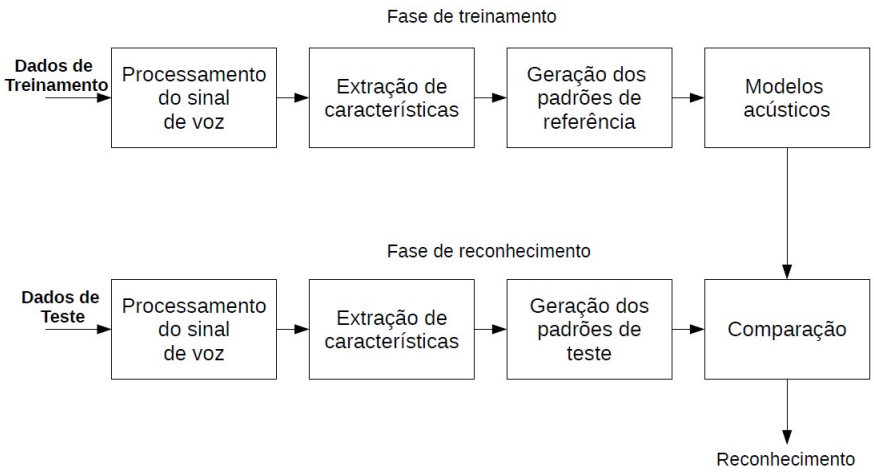

Figura 9: Diagrama de Blocos do Sistema de Reconhecimento de Fala [23].

O reconhecedor de fala usa a Equação 66 para determinar a sequência mais provável de palavras dada a sequência observada na entrada; em que X é uma sequência de vetores acústicos e $\widehat{W}$ a sequência de palavras. Cada palavra é convertida em uma sequência de fonemas e para cada fonema há um modelo estatístico que o corresponde [12] [23] [19] .

Contudo, para a conversão Português-LIBRAS, é necessário o mapeamento de cada letra/palavra reconhecida com seu respectivo sinal em LIBRAS. Essa é a última etapa para a obtenção completa do sistema em que a sequência de fonemas reconhecidos é mapeada para gerar o sinal em LIBRAS que representa. Dessa maneira, as Figuras 10(a) e 10(b) elucidam esse último processo.

Figura 10: (a)Mapeamento letra/LIBRAS; (b)Mapeamento palavra/LIBRAS [Autoria própria].

O HTK (*Hidden Markov Models Toolkit*) é um kit de desenvolvimento para construir e manipular HMM's. A primeira versão do HTK foi disponibilizada em 1989 e foi desenvolvida pelos pesquisadores do Departamento de Engenharia da Universidade de Cambridge [20].

O kit é composto por um conjunto de módulos e ferramentas com código fonte escrito em linguagem C. Esse conjunto foi desenvolvido para aplicações em reconhecimento automático de fala, mas também pode ser utilizado para outras aplicações como síntese de fala, reconhecimento automático de caracteres, dentre outras. Atualmente é uma das ferramentas mais divulgadas e utilizadas por permitir uma fácil integração com outras linguagens, como a linguagem de programação C.

Apesar do kit HTK com os módulos prontos, todas as etapas foram desenvolvidas e replicadas pela linguagem Python; exceto pela última etapa de decodificação. Dessa maneira, este trabalho apresenta as etapas do reconhecedor e as exemplifica em cada passo por meio de uma locução como demonstração dos resultados obtidos. As etapas para o desenvolvimento do reconhecedor são:

Preparação dos Dados e Pré-ênfase:

Nessa primeira etapa de desenvolvimento do reconhecedor de fala, é necessário a obtenção de um banco de dados. Esse banco foi obtido do laboratório de processamento de sinais (LaPS) da UFPA. A partir dele, foram separadas 600 amostras de sinal de voz para o treinamento e teste. Além dos sinais de voz, separou-se os arquivos de texto de todas

as amostras para a criação do dicionário fonético [11].

Contudo, o sinal de voz precisa ser pré-enfatizado para acentuar as frequências mais altas do sinal de voz e tornar o espectro do sinal mais plano. Dessa forma, o sinal passa por um filtro de primeira ordem $L(z)$. Usualmente, os valores de a_p (fator de pré-ênfase) estão em torno de um. [32] [11] [23].

$$L(z) = 1 - a_p z^{-1}. \tag{69}$$

Portanto, a pré-ênfase é dada pela Fórmula 70 que relaciona a saída da pré-ênfase $s_p(n)$ com a entrada $s(n)$ do sinal original.

$$s_p(n) = s(n) - 0.97s(n-1). \tag{70}$$

Segmentação do Sinal de Voz

Os sinais de voz são não-estacionários, ou seja, suas características mudam ao longo do tempo. Por isso, é necessário analisar o sinal em janelas tão pequenas que sejam suficientes para considerar o sinal como estacionário nessa janela. Dessa forma, analisa-se o sinal em uma série de janelas curtas e sobrepostas.

No reconhecimento de fala, é comum o uso de janelas de comprimento de 25 milissegundos e com uma sobreposição de 10 milissegundos. Isso corresponde a uma taxa de 100 quadros por segundo.

Dado que precisa-se extrair um pedaço de um longo sinal contínuo, é importante tomar cuidado com os efeitos de borda. Para sanar isso, utilizamos uma janela de Hamming para a segmentação do sinal por se mostrar mais eficiente que as demais janelas de segmentação [34] [23]. Definida pela Equação 71:

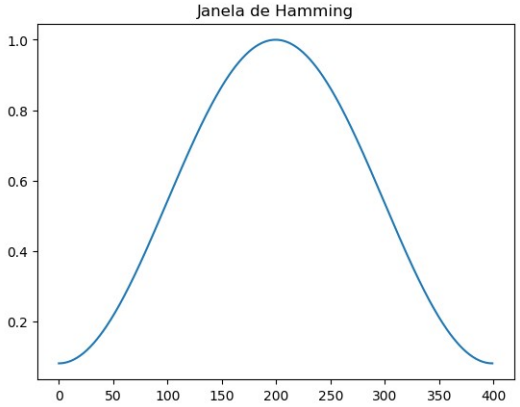

Figura 11: Janela de Hamming [Autoria própria].

$$J(n) = \begin{cases} 0,54 - 0,46cos[2\pi n/(N_A - 1)], & 0 \leq n \leq N_A - 1 \\ 0, & \text{caso contrário.} \end{cases} \tag{71}$$

Extração de Características

Nessa etapa é necessário a extração dos coeficientes mel-cepstrais de todos os arquivos de áudio. A importância da extração destes coeficientes se aplica pela capacidade de sintetização das principais características que definem o sinal de voz [20] [21] [16].

Como ilustração da etapa de extração de características, é ilustrado os resultados obtidos de cada etapa a partir de uma locução do banco de locuções utilizado neste trabalho.

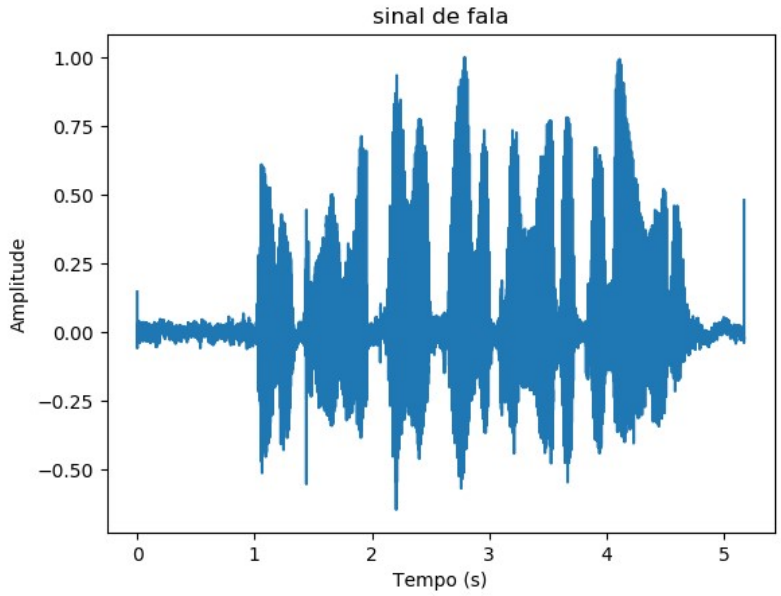

Figura 12: Sinal de fala de uma locução [Autoria própria].

Para cada quadro do sinal, aplica-se uma FFT (*Fast Fourier Transform*) para obter o especto em frequência que passa por um conjunto de filtros triangulares na escala Mel, esse por sua vez possibilita uma melhor representação do comportamento do ouvido humano [25] [23].

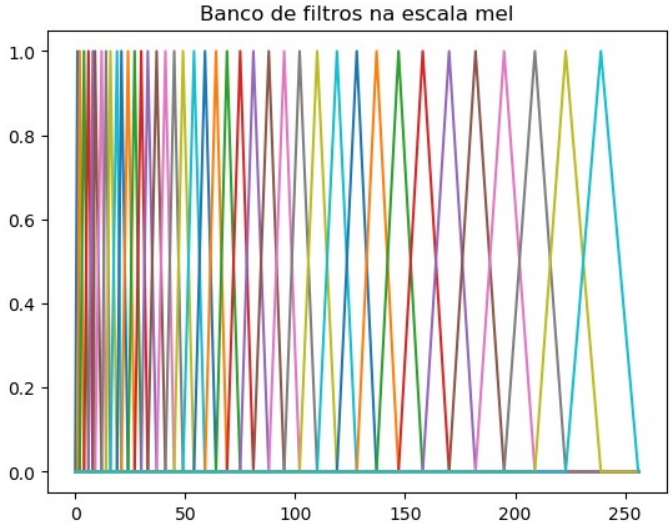

Figura 13: Banco de filtros mel [Autoria própria].

Além disso, realiza-se a compressão logarítmica e então diminui a correlação entre os vetores de parâmetros com o uso de uma DCT (*Discrete Cosine Transform*). Na saída dessa etapa, obtém-se os coeficientes mel cepstrais, dentre os quais, seleciona-se os 12 primeiros juntos com a energia do sinal para a formação de um vetor com 13 componentes em cada quadro [23] [19].

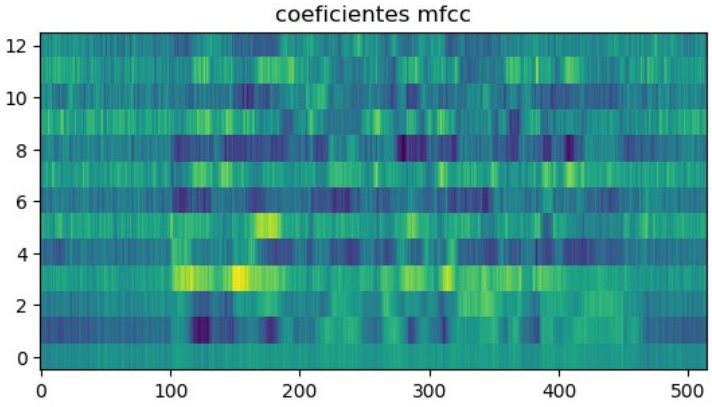

Figura 14: Coeficientes mfcc [Autoria própria].

A seguir, acrescenta-se aos coeficientes MFCC's às suas derivadas de primeira ordem que representam a velocidade de variação do espectro mel-cepstral e as derivadas de segunda ordem que representam a sua aceleração. Ao fim desse processo, obtém-se um vetor de 39 parâmetros por janela como ilustrado pela Figura 15 [23] [19].

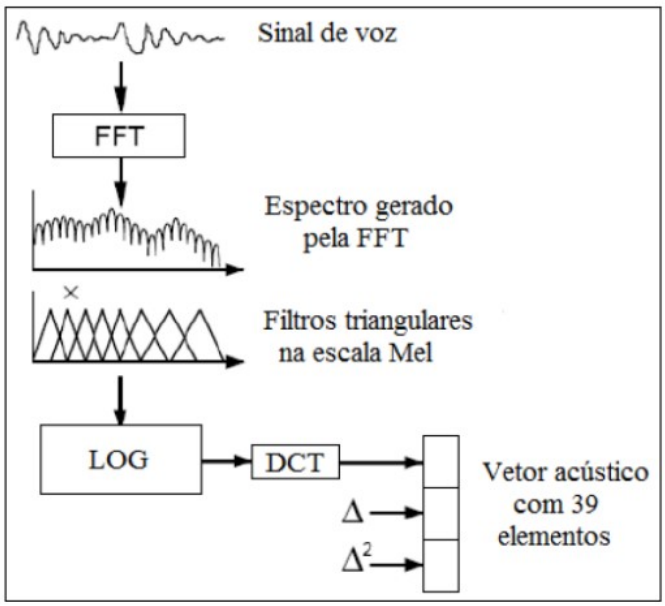

Figura 15: Processador Baseado em MFCCs [19].

Treinamento do Modelo Acústico

Com isso, o algoritmo cria um modelo matemático que utiliza das características acústicas extraídas do sinal. Esse modelo representa cada tipo de segmento da fala, em que a princípio, é utilizado a nível fonético. No treinamento é possível generalizar a descrição

de um fonema desde que se observe as fontes de variabilidades do sinal de voz como algo a ser tratado. Por isso, há a necessidade de que haja um grande conjunto de locuções com a maior variabilidade possível para criar um modelo genérico que mais se aproxime da realidade [23] [20].

Dada uma sequência de palavras W o modelo calcula a verossimilhança com a sequência de vetores X dada; P(X|W) [20].

Na literatura, é comum ver que a modelagem acústica para cada fonema ser representado por um HMM (monofone) é obtida a partir dos coeficientes mel-cepstrais. Dessa forma, pode-se obter um HMM composto pela união dos modelos de fonemas a fim de formar palavras [23] [19].

Para cada fonema, tem-se um HMM com três estados emissores. A combinação desses HMM's geram as palavras que, por sua vez, combinados geram frases; assim como ilustrado na Figura 5.2.

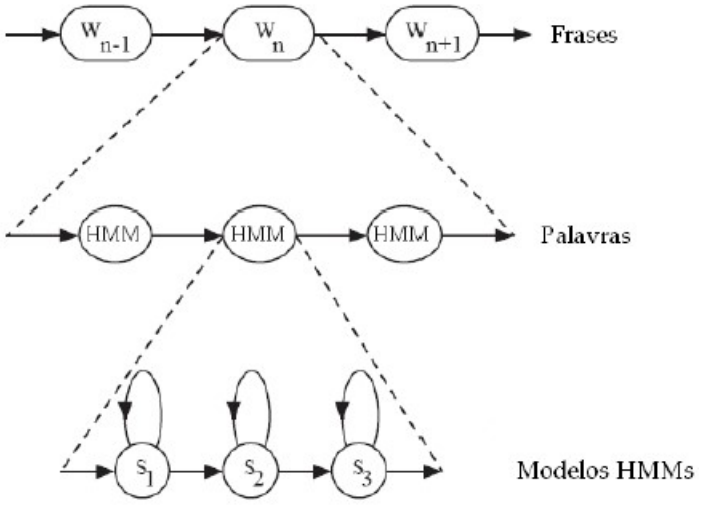

Figura 16: Modelos de HMM concatenados para formar palavras e frases [17].

Tendo em vista a modelagem da fala contínua, é importante considerar o contexto ao qual os fonemas estão sendo pronunciados. Dessa maneira, emprega-se a modelagem a partir de trifones para que cada fone também considere os fonemas à sua direita e à sua esquerda [23] [19].

5.3 Decodificador

Essa última etapa do reconhecimento tem por objetivo transcrever as amostras do sinal de voz em texto. Para isso, é necessário um dicionário e uma rede de fonemas. Enquanto que o dicionário realiza o mapeamento entre fonemas a rede aponta as possíveis transcrições do decodificador [23].

A decodificação de *Viterbi* é um algoritmo que, no espaço de estados, busca a sequência de estados que melhor modele a fala; dado que esse espaço de estados é formado a partir da concatenação dos HMMs. Dessa forma, o algoritmo é uma ferramenta que pro-

cessa cada segmento da fala de forma síncrona [19].

A seguir, é apresentado um exemplo de reconhecimento implementado em parte de uma locução. Na primeira e segunda colunas tem-se o tempo de duração do respectivo fonema que está localizado na terceira coluna seguido da última coluna que apresenta o cálculo da verossimilhança.

Tempo I (μs)	Tempo F (μs)	Transcrição Fonética	Verossimilhança
000000	330000	sil	1204.107300
330000	360000	p	183.064026
360000	410000	e	344.225586
410000	480000	s	473.969818
480000	530000	k	338.399200
530000	600000	i	487.948212
600000	670000	z	436.364166
670000	710000	a	279.523407
710000	750000	sp	254.880325
750000	800000	e	329.080536
800000	830000	u	211.193619
830000	860000	m	190.021927
860000	900000	a	277.999512
900000	950000	k	324.678223
950000	990000	o	302.956573
990000	1020000	i	249.031509
1020000	1060000	z	267.999695
1060000	1090000	a	222.845932
1090000	1090000	sp	0.191865

Tabela 1: Segmentação Automática de uma locução [Autoria própria].

Na etapa de avaliação dos resultados, o decodificador faz a análise dos resultados do reconhecimento de fonemas por meio de um alinhamento forçado e gera uma taxa de erro de fonemas; WER(*Word Error Rate*), que está definida pela fórmula 72 [20] [25].

$$WER = \frac{S_s + I + D_s}{N_s}, \tag{72}$$

tem-se que N_s é o número total de palavras na sequência de teste e S_s, I e D_s são, respectivamente, o número total de erros por substituição (*substitution*), inserção (*insertion*) e supressão (*deletion*) na sequência reconhecida [20].

6 Resultados e discussões

O reconhecedor de fala é implementado em duas etapas. A primeira consiste na etapa de treinamento, em que os modelos probabilísticos são treinados a partir de um banco de dados com locuções de diferentes oradores. O banco utilizado possui 600 frases foneticamente balanceadas composto por locuções de homens e mulheres. Na segunda etapa, verifica-se a taxa de reconhecimento desse sistema por meio de testes, ao qual gerou uma taxa de 73,5% de frases reconhecidas utilizando modelos acústicos de monofones. Após

gerados os resultados do reconhecimento de fala, implementa-se o algoritmo de mapeamento que buscará o sinal em LIBRAS que corresponde àquele sinal de fala encontrado na entrada do sistema. O sistema proposto trabalha com a possibilidade da palavra reconhecida estar contida ou não no dicionário de sinais em LIBRAS. Dessa forma, obtém-se a conversão letra/sinal ilustrada pela Figura 17 e a conversão palavra/sinal ilustrada pela Figura 18.

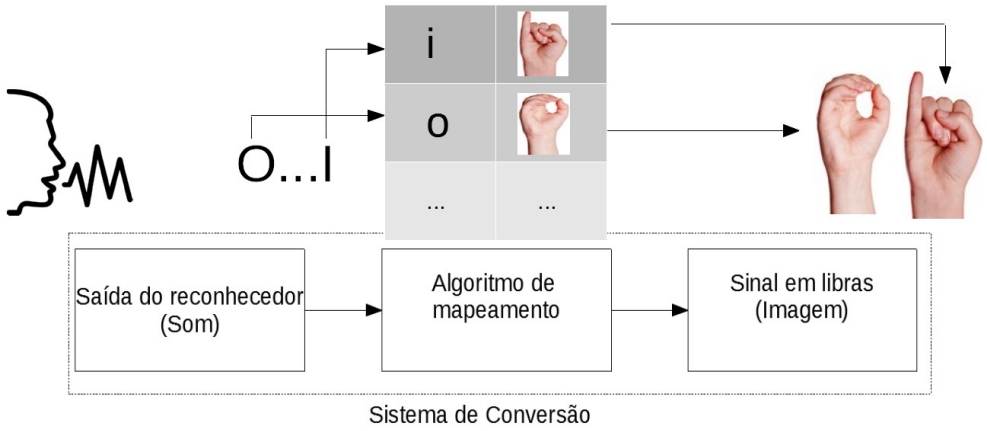

Figura 17: Sistema de Conversão Português-LIBRAS (letra/sinal) [Autoria própria].

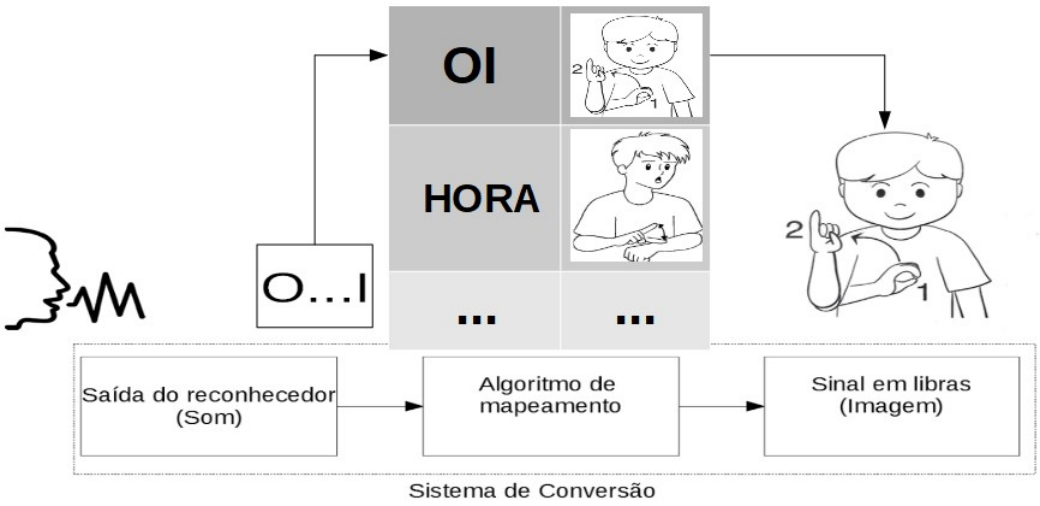

Figura 18: Sistema de Conversão Português-LIBRAS (palavra/sinal) [Autoria própria].

O algoritmo de mapeamento segue conforme ilustrado na Figura 19.

Dada a palavra obtida do reconhecimento de fala, verifica-se sua correspondência ao dicionário. Se essa correspondência existir, realiza-se uma associação palavra/sinal em LIBRAS e declara-se a palavra mapeada com sucesso. Caso a palavra não se encontre no dicionário, verifica-se a possibilidade de haver apenas caracteres que representem letras do alfabeto Português. Quando isso ocorre, garante-se que a palavra não contém erros de transcrição e a mesma é segmentada, letra a letra, e associada a uma imagem do sinal do alfabeto em LIBRAS que a corresponde. Ao final dessa etapa, obtém-se um conjunto de letras que, sequencialmente identificadas por seu sinal em LIBRAS, gera a palavra mapeada. Quando a mesma palavra apresenta caracteres que não correspondem a letras, classifica-se como ininteligível pois a mesma não mapearia uma palavra que fosse

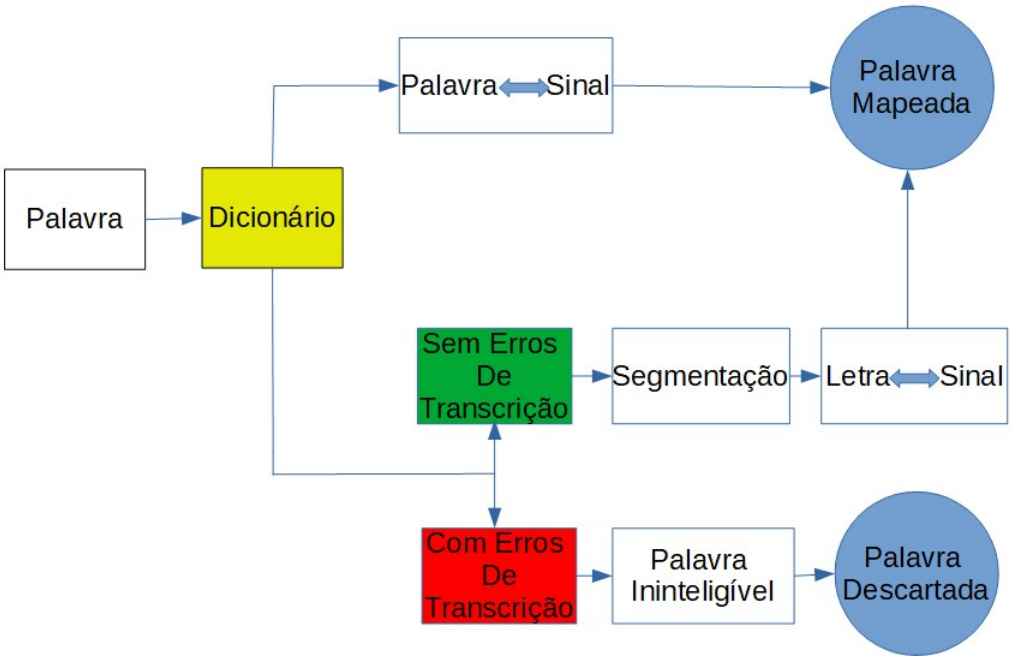

Figura 19: Algoritmo de mapeamento [Autoria própria].

facilmente identificada por surdos. Logo, essa palavra é descartada do mapeamento.

Dado o processamento do sinal de fala de um locutor ter gerado um arquivo com a transcrição fonética, tempo inicial e final de cada fonema e seu cálculo de verossimilhança, considere como exemplo de conversão a palavra 'pesquisa' gerada pela saída do decodificador na Tabela 1.

Considerando que essa palavra foi bem identificada pelo reconhecedor, aplicada como entrada no algoritmo de mapeamento, identificada que não está contida no dicionário e gerada suas respectivas imagens em LIBRAS, pode-se observar a seguinte sequência de imagens:

Figura 20: Mapeamento da palavra pesquisa [Autoria própria].

Assim como ilustrado com a palavra 'pesquisa', toda a frase foi mapeada com os sinais em Libras correspondentes. Esse processo foi aplicado em todas as 600 frases do banco utilizado.

Considere a palavra 'situações' na saída do decodificador, segundo a Tabela 6.

Nesse exemplo, houve uma falha na identificação dos fonemas 's' e '$o\sim$'. O erro foi gerado na etapa de reconhecimento de fala, em que não foi possível obter uma conversão grafema/fonema. Por isso os modelos acústicos para esses fonemas devem ser atualizados para corrigir suas transcrições. É interessante salientar que esse mesmo problema ocorreu, principalmente, nas palavras cujas vogais são nasais ($a\sim$, $e\sim$, $i\sim$, $o\sim$, $u\sim$). Como resultado de um mapeamento falho, obtém-se a seguinte sequência de imagens para a representação da palavra 'situações':

Tempo I (μs)	Tempo F (μs)	Transcrição Fonética	Verossimilhança
560000	590000	s	214.848953
590000	620000	i	208.150284
620000	660000	t	273.837982
660000	690000	u	248.759689
690000	730000	a	313.365295
730000	790000	s	402.475342
790000	840000	o$\sim$	384.243317
840000	890000	i	371.638611
890000	930000	s	275.722168

Tabela 2: Segmentação Automática da palavra 'situações' [Autoria própria].

Figura 21: Mapeamento da palavra 'situações' [Autoria própria].

Por esse motivo, palavras com quaisquer erros em sua decodificação, em um ou mais fonemas, foram excluídas do mapeamento por gerarem palavras ininteligíveis na saída do algoritmo de mapeamento.

Considere como exemplo de conversão palavra/sinal a palavra 'hora' gerada pela saída do decodificador de acordo com a Tabela 3, obteve-se:

Tempo I (μs)	Tempo F (μs)	Transcrição Fonética	Verossimilhança
151000	163000	O	724.217407
163000	166000	r	179.849823
166000	170000	a	260.728516

Tabela 3: Segmentação Automática da palavra 'hora' [Autoria própria].

Dado que a palavra 'hora' foi identificada com sucesso pelo dicionário, a mesma foi associada a um sinal em LIBRAS que corresponde a palavra por inteiro e não a cada letra formada pela palavra como ilustra a Figura 22.

Figura 22: Sinal em LIBRAS da palavra 'hora' [Autoria própria].

Dessa maneira, dada a Figura 23, é possível observar a quantidade de palavras que se trabalhou ao longo das 600 frases bem como o número de palavras cujo os fonemas

foram bem reconhecidos e portanto bem mapeados e também a quantidade de palavras que houve algum fonema com má identificação e portanto deixou a palavra ininteligível para o mapeamento em LIBRAS. Portanto, é possível observar uma taxa de mapeamento de 89,59%.

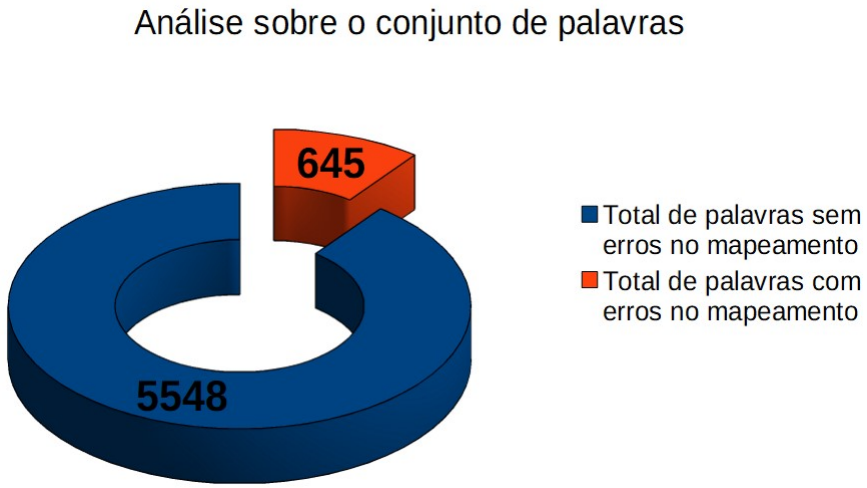

Figura 23: Avaliação das palavras no mapeamento [Autoria própria].

7 Conclusões

Este trabalho apresenta uma nova proposta de conversão entre línguas orais e gesto-visual. Dado que o foco dos conversores convencionais se dá apenas pela conversão entre línguas orais.

Visto a taxa de 73,5% no reconhecimento de fala a partir de monofones, o resultado de 89,59% para o mapeamento de sinais de fala em sinais da LIBRAS se mostraram satisfatórios.

8 Perspectivas de futuros trabalhos

Devido a importância da inclusão da comunidade surda na integração da sociedade nos diversos meios de comunicação, este trabalho gera uma perspectiva de nova ferramenta a ser utilizada em diversos ambientes tais como restaurantes, aeroportos e shoppings center; onde a comunicação rápida e simples entre surdos e ouvintes se faz necessária.

Porém é preciso melhorar as técnicas de reconhecimento de fala, dado que o presente trabalho não considera o contexto em que os fonemas estão dispostos nas palavras.

Por isso é preciso que haja uma maior investigação das técnicas de conversão de idiomas a fim de aproximar o desempenho do conversor de idiomas proposto dos conversores convencionais.

A fim de melhorar o sistema de conversão para a utilização de sinais que contemplem todo o contexto da frase em LIBRAS, é possível aprimorar o desempenho do sistema por meio de técnicas mais avançadas; como a utilização de trifones ao invés de monofones.

Pois aqueles são mais robustos e também consideram o contexto em que os fonemas estão inseridos e, portanto, leva em consideração os fonemas que ocorrem antes e depois de seu fonema central.

Figura 24: Mãos que Falam [35].

9 Outras atividades

•Curso de linguagem de programação (Python) na plataforma online *Udemy.*

•Curso de visualização de dados utilizando a linguagem de programação (Python) na plataforma online *Udemy.*

•Curso de *Data Science: Probability* utilizando a linguagem de programação (Python) na plataforma online *edX.*

•Minicurso de Redação Científica e Plágio Acadêmico com esclarecimento sobre a construção da escrita de textos acadêmicos, bem como a análise de textos classificados como plágio;

•Participação no 1º Encontro de Comunicações e Processamento de Sinais - LABCOM - DEL -UFS;

10 Referências bibliográficas

Referências

[1] W. Altschuler, "Are you a polyglot? here's how to connect through language." https://www.forbes.com/sites/wendyaltschuler/2020/04/22/are-you-a-polyglot-heres-how-to-connect-through-language/6dbe935892c0, 2020. Acesso em 9-julho-2020.

[2] G. do Brasil, "Cidadania e justiça." <http://www.brasil.gov.br/cidadania-e-justica/2016/09/apesar-de-avancos-surdos-ainda-enfrentam-barreiras-de-acessibilidade>, 2016. Acesso em 12-julho-2019.

[3] S. L. do N. C. Costa, *Análise Acústica, Baseada no Modelo Linear de Produção da Fala, para Discriminação de Vozes Patológicas.* Doutorado em ciências no domínio da engenharia elétrica., Universidade Federal de Campina Grande, Paraíba, 2008.

[4] L. J. Raphael, G. J. Borden, and K. S. Harris, *Speech Science Primer*. Lippincott Williams Wilkins, 2011.

[5] H. P. Barbosa, "Aplicação do ppm ao reconhecimento de padrões vocais patológicos," mestrado em ciência da computação, Universidade Federal de Campina Grande, Campina Grande, 2013.

[6] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. John Wiley and Sons, 2003.

[7] R. B. Rocha, "Sistema de segmentação de fala baseado na observação do pitch," jul. 2014.

[8] A. M. Selmini, *Sistema Baseado em Regras para o Refinamento da Segmentação Automática de Fala*. Doutorado em engenharia elétrica, Universidade Estadual de Campinas Faculdade de Engenharia Elétrica e de Computação, Campinas, 2008.

[9] R. D. R. Fagundes and I. Sanches, "Uma nova abordagem foneticofonologica em sistemas de reconhecimento de fala espontânea," dez. 2003. Edição Especial.

[10] T. R. G. Landim, "Sistema de comandos e identificação da voz," 2017.

[11] J. M. Fechine, *Reconhecimento Automático de Identidade Vocal Utilizando Modelagem Híbrida: Paramétrica e Estatística*. Doutorado em engenharia elétrica, Universidade Federal da Paraíba, Campina Grande, 2000.

[12] L. F. M. P. Coelho, "Etiquetagem automática de sinais de fala segmentação e classificação fonética," mestre em engenharia eletrotécnica e de computadores, Faculdade de Engenharia da Universidade do Porto, Porto - Portugal, 2005.

[13] M. Nilsson and M. Ejnarsson, "Speech recognition using hidden markov model," master of science in electrical engineering, Blenking Institute of Technology, Ronneby, 2002.

[14] V. L. Latsch, "Construção de banco de unidades para síntese de fala por concatenação no domínio temporal," mestre em ciências em engenharia elétrica, Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2005.

[15] A. C. da Silva, F. M. E. Camilo, and T. V. D. Ferraz, "Reconhecimento de fala dependente de locutor utilizando redes neurais artificias," 2013.

[16] C. A. Ynoguti, *Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov*. Doutorado em engenharia elétrica, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, 1999.

[17] C. P. A. da Silva, "Um software de reconhecimento de voz para português brasileiro," mestrado em engenharia elétrica, Universidade Federal do Pará, Belém, 2010.

[18] J. da Rosa Júnior, "Reconhecimento automático de emoções através da voz," 2017.

[19] R. T. Tevah, "Implementação de um sistema de reconhecimento de fala contínua com amplo vocabulário para o português brasileiro," mestrado em ciências em engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

[20] C. U. E. Department, *The HTK Book*. 2009.

[21] E. D. S. Paranaguá, "Reconhecimento de locutores utilizando modelos de markov escondidos contínuos," mestrado em ciências em engenharia, Instituto Militar de Engenharia, Rio de Janeiro, 1997.

[22] S. V. Vaseghi, *Advanced Digital Signal Process and Noise Reduction.* Wiley, 2006.

[23] R. B. Rocha, "Desenvolvimento de um codificador de voz pessoal de baixa taxa baseado em modelos de markov escondidos," mestrado em engenharia elétrica, Universidade Federal de Campina Grande - UFPB, Campina Grande - PB, 2012.

[24] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," jan. 1988.

[25] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* PTR Prentice-Hall, 1993.

[26] P. Dymarski, *Hidden Markov Models, Theory and Applications.* InTech, 2011.

[27] A. M. da Cunha, *Movimentos de Cabeça guiados pela Voz.* Tese de doutorado, IMPA, Rio de Janeiro, 2009.

[28] L. da S. Espindola, "Um estudo sobre modelos ocultos de markov," 2009.

[29] R. C. Alamino, *Aprendizado em Modelos de Markov com Variáveis de Estado Escondidas.* Doutorado em ciências, Universidade de São Paulo, São Paulo, 2005.

[30] B. L. Cardoso, "Sistema de reconhecimento de fala baseado em pocketsphinx para acessibilidade em android," mestre em engenharia elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

[31] L. Debatin, "Desenvolvimento de uma ferramenta para análise de desempenho do reconhecimento off-line de voz contínuo em dispositivos móveis," mestre em computação aplicada, Universidade do Vale do Itajaí, Itajaí, 2019.

[32] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, "Discrete time processing of speech signals," 1993.

[33] L. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Prentice Hall, New Jersey, 1978.

[34] J. Picone, "Signal modeling techniques in speech recognition," 1993.

[35] D. Vanderlei, "Mãos que falam." http://nasmaosdafala.blogspot.com/, 2012. Acesso em 12-julho-2020.