UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

Relatório Final - MS777

Modelagem matemático/probabilística dos módulos acústicos e de linguagem de sistemas de reconhecimento de fala.

Rodolfo Rodrigues de Araujo 082720

Orientador: Dr. Edmilson da Silva Morais

Campinas, 03 de Julho de 2010

1. Resumo

Este trabalho consistiu no treinamento e avaliação de um sistema de reconhecimento de fala.

O desenvolvimento desse trabalho ocorreu junto a Vocalize, empresa formada por ex-alunos de doutorado da FEEC e IEL. A Vocalize permitiu a utilização de Corpora de áudio e texto, as respectivas transcrições fonéticas além do suporte técnico e ferramental necessário.

2. Objetivos

O principal objetivo desse trabalho foi o estudo da modelagem matemática envolvida no modelo acústico de um sistema de reconhecimento de fala.

3. Desenvolvimento

3.1 Introdução

Apesar de as primeiras pesquisas na área de Reconhecimento Automático de Fala (RAF) datarem da década de 50, somente nos últimos anos esses sistemas passaram a apresentar resultados considerados de alta qualidade. Esses recentes avanços nos sistemas RAF devem-se à elevada disponibilidade de Corpora de texto e fala e ao aumento da capacidade de processamento dos computadores atuais, os quais têm permitido o desenvolvimento de algoritmos cada vez mais complexos.

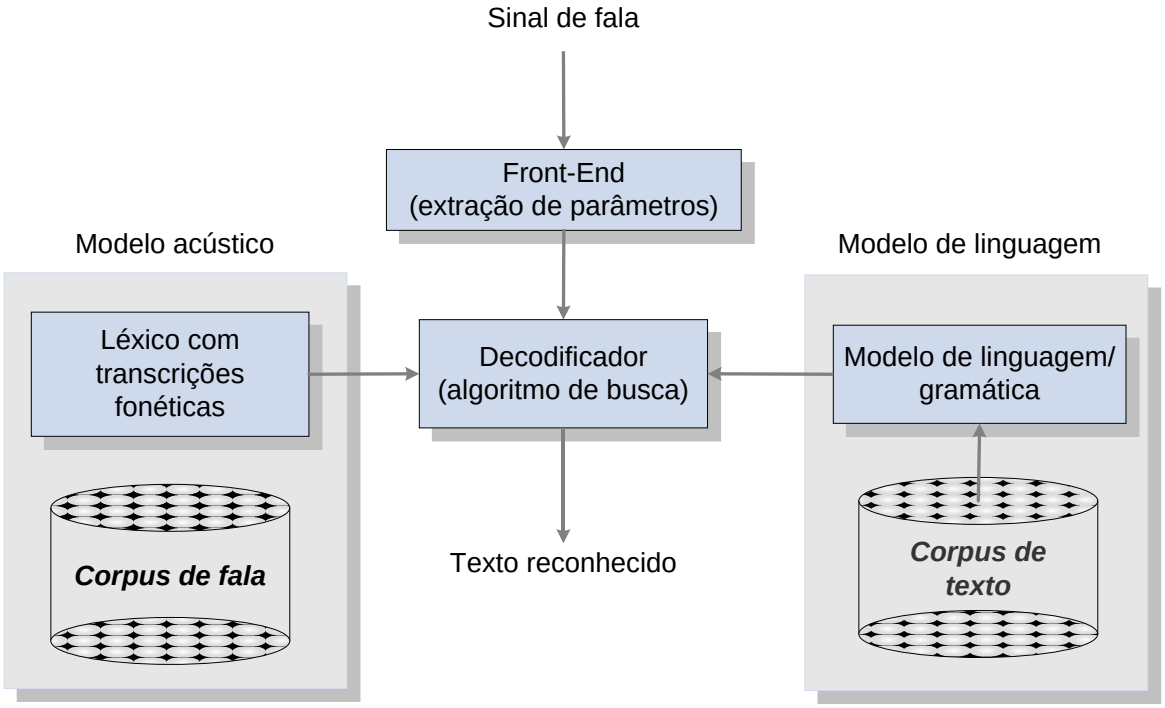
O *estado-da-arte* dos atuais sistemas RAF [1,3] emprega técnicas probabilísticas tais como: modelos ocultos de Markov – HMM (*Hidden Markov Models*) [1,2,3], árvores de classificação [4] e métodos de busca baseados no algoritmo de *Viterbi* [1,2]. Essas técnicas probabilísticas são utilizadas para o treinamento e avaliação de funções probabilísticas capazes de mapear amostras de um sinal de fala – representadas através de uma seqüência de vetores acústicos – no espaço de possíveis sentenças (palavras ou frases). Essas funções probabilísticas são projetadas/treinadas para serem capazes de modelar as duas principais fontes de variabilidade da fala: a *variabilidade acústica* e a *variabilidade temporal*.

O objetivo de um sistema RAF consiste em estimar, durante a etapa de treinamento, os parâmetros dos modelos HMM e utilizar, durante a etapa de reconhecimento, a seguinte função probabilística, $P(M \mid \mathbf{X}, \Theta)$, onde $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ é uma seqüência de vetores acústicos derivados do sinal de fala a ser reconhecido, $s(n)$, através de um procedimento de **Pré-Processamento,** e sendo M_i $(i = 1,2,\cdots, I_M)$ o conjunto de todas as possíveis sentenças que podem corresponder a $s(n)$ e Θ o conjunto de parâmetros associados aos modelos HMM. Uma vez que uma sentença M pode ser construída a partir da concatenação de palavras $M = \{W_1, W_2, \cdots, W_{N_M}\}$, a tarefa de um sistema RAF também pode ser interpretada como a determinação da seqüência de palavras mais prováveis $\hat{M}$, dada a seqüência de vetores acústicos $\mathbf{X}$ e o conjunto de parâmetros Θ. Se a regra de *Bayes* for utilizada para decompor $P(M \mid \mathbf{X}, \Theta)$, então $\hat{M}$ pode ser determinada a partir da seguinte expressão:

$$\hat{M} = \arg\max_M P(M \mid \mathbf{X}, \Theta) = \arg\max_M \frac{p(\mathbf{X} \mid M, \Theta) \cdot P(M \mid \Theta)}{p(\mathbf{X} \mid \Theta)} \qquad (1)$$

Deve-se observar que o termo $p(\mathbf{X} \mid \Theta)$ independe de M_i e, portanto, não necessita ser calculado. Esta equação mostra que encontrar $\hat{M}$ é equivalente a encontrar a seqüência de palavras que maximiza o produto entre $p(\mathbf{X} \mid M, \Theta)$ e $P(M \mid \Theta)$. Uma vez que durante o reconhecimento o termo $p(\mathbf{X} \mid M, \Theta)$ representa a verossimilhança da seqüência de vetores acústicos $\mathbf{X}$ dada uma seqüência de palavras específica M e o conjunto de parâmetros Θ, esta verossimilhança pode ser determinada a partir de um **Modelo acústico** (empregando-se HMM) para sentença M. O segundo termo $P(M \mid \Theta)$ representa a probabilidade da sentença M dado o conjunto de parâmetros Θ, e esta probabilidade pode ser determinada por um **Modelo de linguagem**.

O processo de determinação de $\hat{M}$ é denominado decodificação e projetos de decodificadores (algoritmos de busca) eficientes são cruciais para a realização prática de sistemas para reconhecimento de fala contínua. Portanto, um sistema RAF pode ser dividido em quatro módulos principais: **(1) Pré-processamento(extração de parâmetros)** , **(2) Modelo Acústico**, **(3) Modelo de Linguagem (Modelo da Língua)** e **(4) Decodificação (Algoritmo de Busca), figura 1**.



3.2 Estudos e Experimentos Realizados Nesse Trabalho

Devido à complexidade de um sistema de reconhecimento de fala contínua, o presente trabalho concentrou-se apenas na análise das técnicas fundamentais para o treinamento dos modelos acústico.

Nas seções seguintes será feito um estudo sobre os temas relacionados a esse tipo de sistema.

3.2.1. Modelagem acústico

3.2.1.1. Fundamentos sobre HMM

HMMs (*Hidden Markov Models*) são máquinas de estados finitas que geram observações discretas (símbolos). A cada unidade de tempo a HMM muda de estado, de acordo com uma distribuição de probabilidade, e então emite um símbolo, de acordo com uma distribuição de probabilidade de emissão do estado corrente.

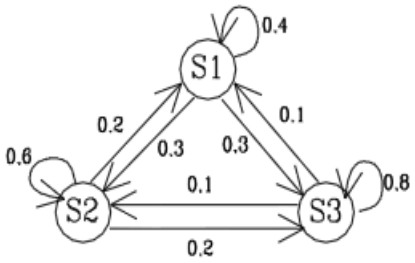


Fig. 2: Máquina de estado representando o modelo

Elementos de uma HMM:

S = $\{1, \ldots, N\}$ é número de estados

$\boldsymbol{V} = \{v_1, v_2, \ldots, v_M\}$ é o número de símbolos

$\boldsymbol{O} = \{o_1, o_2, \ldots, o_T\}$ é a seqüência de observações até o tempo T

A = matriz probabilidade de transição entre os estados

$a_{ij} =$ **probabilidade de transição estando no estado i e passar para o j**

B = probabilidade de emissão de símbolos

$b_j(o_k) =$ **probablidade do estado j emitir o símbolo da observação** o_k

$\pi_i =$ probabilidade inicial

Pode-se calcular, por exemplo, a probabilidade de ocorrer a seqüência de observações ***O*** dado o modelo λ :

$$P(\boldsymbol{O}|\lambda) \; onde \; \lambda = (A, B, \pi) \qquad (3)$$

Os três problemas básicos são:

(1) Dado uma seqüência de observações ***O*** e o modelo λ **,** qual a probabilidade desta seqüência ter sido gerada pelo modelo?

(2) Dado uma seqüência de observações ***O*** e o modelo λ **,** qual a seqüência de estados mais provável?

(3) Dada uma seqüência ou conjunto de seqüência de observações ***O***, de que forma se ajusta os parâmetros do modelo λ de modo a maximizar a probabilidade que ela ocorra?

3.2.3.1.1. Resolução do problema 1

Assumindo $Q = \{q_1, q_2, \ldots, q_T\}$ conhecida, então a probabilidade da seqüência de observações **O** ter sido gerada pelo modelo λ é $P(O|Q,\lambda)$. A probabilidade que ocorra a seqüência de estados **Q** dado o modelo é $P(Q|\lambda)$.

A probabilidade conjunta de ***Q*** e ***O*** é:

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda) \quad (4)$$

A resposta do problema é o resultado da soma de todas as seqüências de estados possíveis da probabilidade conjunta:

$$P(O|\lambda) = \sum_{\forall q} P(O|Q,\lambda)P(Q|\lambda) \quad (5)$$

Esta expressão implica em um número elevado de operações, $(2T-1)N^T + N - 1$. Para contornar tal situação, foi desenvolvido o procedimento *Forward-Backward*.

3.2.1.1.2. Algoritmo *Forward*

Considere a variável *forward* $\alpha_t(i) = P(o_1 o_2 \ldots o_t, q_t = i|\lambda)$ como a probabilidade da observação parcial até o tempo *t* no estado *i* dado o modelo λ . Em outras palavras, probabilidade acumulada considerando todos os caminho que chegam ao estado *i* no tempo *t* emitindo a seqüência de observações até o_t .

Passo 1: será calculado o alfa para os estados iniciais. Deve-se considerado para cada estado a probabilidade inicial de estar no estado e a probabilidade de cada estado emitir a observação o_1 , Figura 3.

➔ Tempo 1

$\alpha_1(1) = \pi_1 b_1(o_1)$

$\alpha_1(2) = \pi_2 b_2(o_1)$

$\alpha_1(3) = \pi_3 b_3(o_1)$

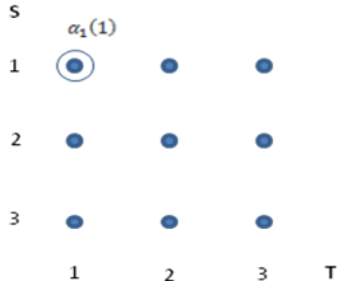


Fig. 3: Máquina de estado representando o primeiro passo

Passo 2: para cada estado subseqüente, deve-se calcular a probabilidade acumulada de todos os caminhos possíveis até o estado corrente, emitindo a observação do tempo *t*.

➔ Tempo 2

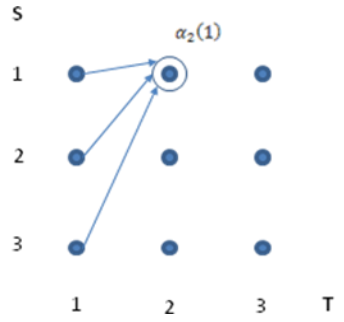


Fig. 4: Máquina de estado representando o segundo passo

$$\alpha_2(1) = [\alpha_1(1)\, a_{11} + \alpha_1(2)\, a_{21} + \alpha_1(3)\, a_{31}]\, b_1(o_2)$$
$$\alpha_2(2) = [\alpha_1(1)\, a_{12} + \alpha_1(2)\, a_{22} + \alpha_1(3)\, a_{32}]\, b_2(o_2)$$
$$\alpha_2(1) = [\alpha_1(1)\, a_{13} + \alpha_1(2)\, a_{23} + \alpha_1(3)\, a_{33}]\, b_3(o_2)$$

➔ Tempo 3

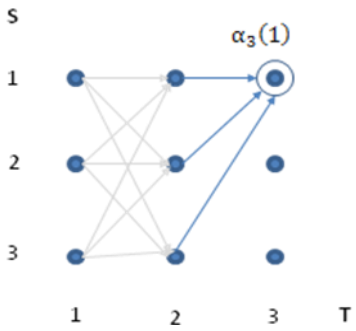


Fig. 5: Máquina de estado representando o terceiro passo

Note que a probabilidade representada por $\alpha_2(i)$ já traz as probabilidades acumuladas anteriormente.

$$\alpha_3(1) = [\alpha_2(1)\, a_{11} + \alpha_2(2)\, a_{21} + \alpha_2(3)\, a_{31}]\, b_1(o_3)$$
$$\alpha_3(2) = [\alpha_2(1)\, a_{12} + \alpha_2(2)\, a_{22} + \alpha_2(3)\, a_{32}]\, b_2(o_3)$$

$$\alpha_3(3) = [\alpha_2(1)\, a_{13} + \alpha_2(2)\, a_{23} + \alpha_2(3)\, a_{33}]\, b_1(o_3)$$

Passo 3: Para saber a probabilidade total basta somas as probabilidades acumuladas até o último estado:

$$P(O|\lambda) = \alpha_3(1) + \alpha_3(2) + \alpha_3(3)$$

Então o Algoritmo *Forward* fica:

1. Inicialização

$$\alpha_1(i) = \pi_i\, b_i(o_1) \quad \text{para} \quad 1 \leq i \leq N \tag{6}$$

2. Indução

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(o_{t+1}) \quad \text{para} \quad 1 \leq t \leq T-1 \quad e \quad 1 \leq j \leq N \tag{7}$$

3. Término:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{8}$$

3.3.1.1.3. Algoritmo *Backward*

De maneira similar ao Forward, nós podemos considerar a variável *backward* $\beta_t(i)$ definida por

$$\beta_t(i) = P(O_{t+1} O_{t+2} \ldots O_T | q_t = i, \lambda) \tag{9}$$

ou seja, a probabilidade parcial da sequência de observações de t+1 até o fim, dado o estado i no tempo t e o modelo λ . Exemplo:

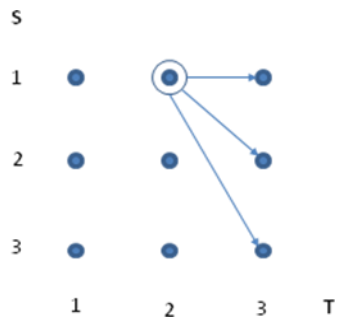


Fig. 6: $\beta_2(1)$ = prob. acumulada de todos os caminhos que partem do estado 1 no tempo 2 emitindo a sequência parcial de símbolos O_3.

O algoritmo:

1. Inicialização

$$B_t(i) = 1, \qquad 1 \leq i \leq N \tag{10}$$

2. Indução

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \; para \; t = T-1, T-2, \ldots, 1 \; e \; 1 \leq i \leq N \tag{11}$$

3. Término

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i) \tag{12}$$

Exemplo**:** Considerando um caso particular onde N = 3 e T = 3, exemplificado a seguir, seguimos os seguintes passos:

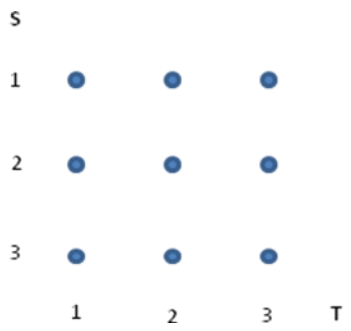


Fig. 7: Caso particular do modelo onde N = 3, T = 3.

Passo 1:

$\beta_3(1) = 1$

$\beta_3(2) = 1$

$\beta_3(3) = 1$

Passo 2:

$$\beta_2(1) = a_{11}b_1(O_3)\beta_3(1) + a_{12}b_2(O_3)\beta_3(2) + a_{13}b_3(O_3)\beta_3(3)$$

$$\beta_2(2) = a_{21}b_1(O_3)\beta_3(1) + a_{22}b_2(O_3)\beta_3(2) + a_{23}b_3(O_3)\beta_3(3)$$

$$\beta_2(3) = a_{31}b_1(O_3)\beta_3(1) + a_{32}b_2(O_3)\beta_3(2) + a_{33}b_3(O_3)\beta_3(3)$$

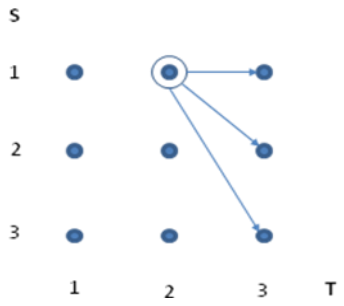


Fig. 8: Cálculo de $\beta_2(1)$.

Passo 3:

$$\beta_1(1) = a_{11}b_1(O_2)\beta_2(1) + a_{12}b_2(O_2)\beta_2(2) + a_{13}b_3(O_2)\beta_2(3)$$

$$\beta_1(2) = a_{12}b_1(O_2)\beta_2(1) + a_{22}b_2(O_2)\beta_2(2) + a_{23}b_3(O_2)\beta_2(3)$$

$$\beta_1(3) = a_{31}b_1(O_2)\beta_2(1) + a_{32}b_2(O_2)\beta_2(2) + a_{33}b_3(O_2)\beta_2(3)$$

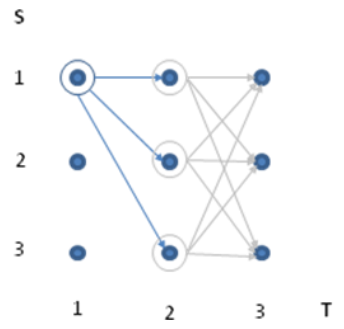


Fig. 9: Cálculo de $\beta_1(1)$.

Passo 4:

$$P(O|\lambda) = \pi_1 b_1(O_1)\beta_1(1) + \pi_2 b_2(O_1)\beta_1(2) + \pi_3 b_3(O_1)\beta_1(3)$$

Considerações:

Note que expandindo o valor de $\beta_1(1)$, temos,

$$\beta_1(1) = a_{11} b_1(O_2)\beta_2(1) + a_{12} b_2(O_2)\beta_2(2) + a_{13} b_3(O_2)\beta_2(3)$$

$$= a_{11} b_1(O_2)[a_{11} b_1(O_3)\beta_3(1) + a_{12} b_2(O_3)\beta_3(2) + a_{13} b_3(O_3)\beta_3(3)]$$

$$+ a_{12} b_2(O_2)[a_{21} b_1(O_3)\beta_3(1) + a_{22} b_2(O_3)\beta_3(2) + a_{23} b_3(O_3)\beta_3(3)]$$

$$+ a_{13} b_3(O_2)[a_{31} b_1(O_3)\beta_3(1) + a_{32} b_2(O_3)\beta_3(2) + a_{33} b_3(O_3)\beta_3(3)]$$

Fazendo a distributiva e observando que $\beta_3(i) = 1, i \in \{1,2,3\}$, segue que,

$$\beta_1(1) = a_{11} b_1(O_2)a_{11} b_1(O_3) + a_{11} b_1(O_2)a_{12} b_2(O_3) + a_{11} b_1(O_2)a_{13} b_3(O_3) + a_{12} b_2(O_2)a_{21} b_1(O_3) + a_{12} b_2(O_2)a_{22} b_2($$

onde cada termo do somatório corresponde a probabilidade de observação das emissões a partir de cada caminho possível de q_1.

3.2.1.1.4. Resolução do problema 2

Para encontrar uma sequência ótima de estados $q = (q_1 q_2 \dots q_T)$ (de acordo com um critério de otimização), dado uma sequência de observações $O = (o_1 o_2 \dots o_T)$, são definidas as seguintes quantidades

$$\delta_t(i) = \max_{T}(q_1 1\ q_1 2\ [\dots q]_t(t-1))\, P[q_1 1\ q_1 2\ [\dots q]_t(t-1)\ [, q]_t t = i, o_1 1\ o_1 2 \dots o \quad (13)$$

Isto é, $\delta_t(i)$ é a maior probabilidade ao longo de um único caminho, até o tempo t, que contabiliza as primeiras t observações e acaba no estado i.

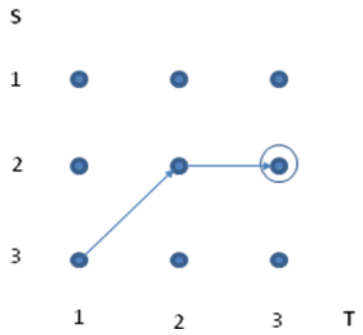


Fig. 10: $\delta_2(2)$ = **probabilidade ao longo do melhor caminho até o estado 2 no tempo 3.**

E também a quantidade $\psi_t(i)$, que representa o estado j exatamente anterior ao estado i, no tempo t, que contabilizou $\delta_t(i)$ e é usada para obter a sequência de estados ótima (*backtracking*). Ou seja,.

$$\psi_t(i) = \arg\max_{1 \le j \le N} \left[\delta_{t-1}(j) a_{ji}\right] \quad (14)$$

Algoritmo:

1. Inicialização

$$\delta_1(i) = \pi_i b_i(o_1) \quad para \quad 1 \le i \le N \quad (15)$$

$$\psi_1(i) = 0 \quad para \quad 1 \le i \le N \quad (16)$$

2. Indução

$$\delta_t(j) = \max_{1 \le i \le N} \left(\delta_{t-1}(i) a_{ij} b_j(o_t)\right) \quad para \quad 2 \le t \le T, 1 \le j \le N \quad (17)$$

$$\psi_t(i) \arg = \max_{1 \le i \le N} \left(\delta_{t-1}(i) a_{ij}\right) \quad para \quad 2 \le t \le T, 1 \le j \le N \quad (18)$$

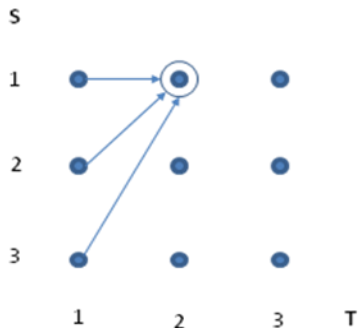


Fig. 11: Cálculo do $\delta_2(1)$. Processo similar ao forward, mas ao invés de acumular as probabilidades parciais anteriores, tira-se o máximo. E guardamos em $\psi_2(1)$ o estado que obteve a probabilidade máxima.

3. Término

$$P^* = \max_{1 \le i \le N} (\delta_t(i)) \quad (19)$$

$$q_T^* \arg\square = \max_{1 \le i \le N} (\delta_t(i)) \quad (20)$$

4. Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \; para \; t = T-1, T-2, \ldots 1. \quad (21)$$

3.2.1.1.5. Resolução do problema 3

Para resolver o problema de estimação utiliza-se o Algoritmo *Baum-Welch* (*Expectation-Maximization*). Esse algoritmo utiliza as variáveis de *Forward-Backward* para a definição das variáveis γ e ξ.

3.2.1.1.5.1. Definição da variável γ

Seja a probabilidade de estar no estado *i* no tempo t dado a seqüência de observações e o modelo $\gamma_t(i) = P(q_t = i | O, \lambda)$. Pela propriedade de *Bayes* tem-se:

$$P(q_t = i | O, \lambda) = (P([O, q]_t = i | , \lambda))/(P(O | \lambda)) \; . \quad (22)$$

Com a resolução do problema 1 sabe-se que $P(O | \lambda) = \Sigma_t(i = 1)^N [P(O, q_t = i | \lambda)]$, então

$$\gamma_t(i) = P(q_t = i | O, \lambda) = (P([O, q]_t = i | , \lambda))/(\Sigma_t(i = 1)^N [P(O, q_t = i | \lambda)]) \; . \quad (23)$$

A probabilidade $P([O, q]_t = i | , \lambda)$ pode ser escrita em termos das variáveis forward-backward:

$$[\alpha_t(i) = P(o_1 \; o_2 \; [\ldots o]_t \; , q]_t = i | \lambda) = P(o_1 \; o_2 \; [\ldots o]_t | \lambda) \; P(q_t = i | \lambda) \quad (24)$$

$$\frac{P(o_{t+1} \; o_{t+2} \; \ldots o_T \; , q_t = i | \lambda))}{P(q_t = i | \lambda)})$$

Logo,

$$\alpha_t(i)\ \beta_t(i) = P(o_1\ o_2\ \ldots\ o_t \mid \lambda)\ P(o_{t+1}\ o_{t+2}\ \ldots\ o_T, q_t = i \mid \lambda) \quad (25)$$

$$\alpha_t(i)\ \beta_t(i) = P(O, q_t = i \mid \lambda)$$

Voltando a variável gama,

$$\gamma_t(i) = (P(O, q_t = i \mid \lambda))/(\Sigma_{i=1}^{N} P(O, q_t = i \mid \lambda)) \quad (26)$$

e sabendo que $\alpha_t(i)\ \beta_t(i) = P(O, q_t = i \mid \lambda)$ chega-se na seguinte expressão:

$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\,\beta_t(i)} \quad (27)$$

3.2.1.1.5.2. Definição da variável ξ

Seja a probabilidade de estando no estado *i* no tempo *t* e no estado j em *t+1* dado o modelo e a seqüência de observação $\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid O, \lambda)$. Pela propriedade de Bayes tem-se:

$$\xi_t(i,j) = (P(q_t = i, q_{t+1} = j, O \mid \lambda))/(P(O \mid \lambda)) \quad (28)$$

A probabilidade $P(q_t = i, q_{t+1} = j, O \mid \lambda)$ pode ser calculada em função das variáveis *Forward-Backward*. Considere a figura abaixo representando a transição entre dois estágios em um tempo intermediário:

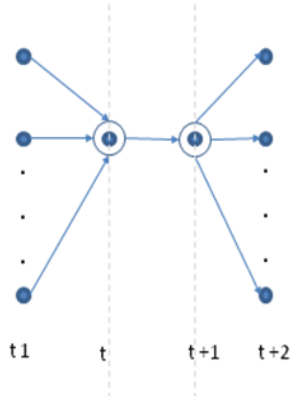


Fig. 12: Máquina de estado representando a transições de estados

O valor de $P(q_t = i, q_{(t+1)} = j, O|\lambda)$ é a probabilidade acumulada até *t* junto a probabilidade de transição, probabilidade de emissão em *t+1* e probabilidade acumulada após *t+1*:

$$P(q_t = i, q_{(t+1)} = j, O|\lambda) = \alpha_t(i)\,[a_{ij}\; b_j(o_{(t+1)})\; \beta]_{(t+1)}(j) \quad (29)$$

Para a transição de dois estados *i* e *j* a expressão da probabilidade é $P(q_t = i, q_{(t+1)} = j, O|\lambda) = \alpha_t(i)\,[a_{ij}\; b_j(o_{(t+1)})\; \beta]_{(t+1)}(j)$, então para todas as possibilidades de *i* e *j* a probabilidade dado o modelo é:

$$P(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j) \quad (30)$$

Voltando a expressão de ξ ,

$$\xi_t(i,j) = \frac{\alpha_t(j)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)} \quad (31)$$

Tomando as definições de γ e ξ:

$$\gamma_t(i) = P(q_t = i|O, \lambda)$$

$$\xi_t(i,j) = P(q_t = i, q_{(t+1)} = j|O, \lambda)$$

Se γ é a probabilidade de no tempo *t* estar no estado *i* e ξ é a probabilidade de estar no estado *i* no tempo *t* e no estado j em *t+1,* então pode-se escrever uma em função da outra. γ é igual a ξ cobrindo todas as possibilidades de transição de *i* para *j*:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \quad (32)$$

3.2.3.1.5.3. Algoritmo *Expectation-Maximization*

Considere as seguintes definições:

$$\text{Nº esperado de transições do estado } i \text{ em } O = \sum_{t=1}^{T-1} \gamma_t(i) \quad (33)$$

$$\text{Nº esperado de transições do estado } i \text{ para o j em } O = \sum_{jt=1}^{T-1} \xi_t(i,j) \tag{34}$$

Utilizando as expressões acima e conceitos de contagem de eventos e ocorrências, pode-se ser dado o método de re-estimação de parâmetro do HMM:

$$\bar{\pi}_j = \text{freqüência esperada (número de vezes) no estado } i \text{ no tempo t} \tag{35}$$

$$\bar{a}_{ij} = \frac{\text{número esperado de transições do estado } i \text{ para o } j}{\text{número esperado de transições do estado } i} = \frac{\sum_{jt=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{36}$$

$$\bar{b}_i(k) = \frac{\text{número esperado de vezes no estado } i \text{ observando o símbolo } v_k}{\text{número esperado de vezes no estado } j} \tag{37}$$

$$\text{nº esp. de vezes no estado } i \text{ observando o símbolo } v_k = \sum_{t=1}^{T} P(q_t = j, o_t = v_k | O, \lambda) \tag{38}$$

(39)

$$\left[\sum_{\substack{t=1 \\ \text{s.a.: } o_t = v_k}}^{T} P(q_t = j | O, \lambda) \right] = \sum_{\underbrace{t=1}_{\text{s.a.: } o_t = v_k}}^{T} \gamma_t(i))$$

$$\bar{b}_i(k) = \frac{\underbrace{\sum_{t=1}^{T} \gamma_t(i)}_{\text{s.a.: } o_t = v_k}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{40}$$

4. **Experimento**

O trabalho realizado concentrou-se na utilização da ferramenta HTK(HMM Tool Kit), utilizando-a nas etapas e treinamento e reconhecimento, junto com a ferramenta SRILM(SRI Language Model) responsável por gerar os modelos de linguagem.

HTK

O HTK é um conjunto de ferramentas desenvolvido pelo Departamento de Engenharia da Universidade de Cambridge na Inglaterra - CUED (*Cambridge University Engineering Department*) utilizado para construir e manipular modelos ocultos de Markov – HMMs (*Hidden Markov Models*). O HTK é usado para construir sistemas de processamento de fala baseados em HMMs, com foco principal em sistemas de reconhecimento de fala, mas é

utilizado também em diversas outras aplicações como em pesquisas de síntese de voz, processamento de linguagem e seqüenciamento de DNA, por exemplo. Por ser distribuído livremente junto com o seu código-fonte e possui uma licença pouco restritiva, que permite o uso do pacote para pesquisas e para desenvolvimento de aplicações, mas não permite que produtos comerciais contenham partes de seu código. O *software* é utilizado amplamente tanto no meio acadêmico quanto em empresas comerciais (para o desenvolvimento de protótipos iniciais).

O HTK consiste basicamente em um conjunto de módulos e ferramentas escritos na linguagem C e, portanto, disponíveis para uso em diversos sistemas operacionais, sendo que as ferramentas cumprem diversas funções como treinamento de HMMs, teste e análise de sistemas de reconhecimento. Os sistemas de RAF construídos através do HTK funcionam da seguinte forma: primeiramente, as ferramentas de treinamento do HTK são utilizadas para estimar os parâmetros dos HMMs com o uso de sinais de fala conhecidos em conjunto com suas transcrições fonéticas. Em seguida, o sistema recebe como entrada sinais de fala desconhecidos e gera como saída os textos referentes às entradas. O *software* possui uma página *web* [5] com várias informações e onde se pode obtê-lo, assim como um manual conhecido como "*HTK Book*" que contém vasta documentação sobre todos os aspectos do pacote.

SRILM

O SRILM é um *toolkit* utilizado para construir modelos de linguagem estatísticos para aplicações em reconhecimento de fala, rotulagem estatística, etiquetadores morfossintáticos e segmentação – está em desenvolvimento desde 1995 pelo SRI *International*. O *software* dá ênfase principalmente aos modelos *N-grams* e consiste dos seguintes componentes: um conjunto de bibliotecas de classes escritas em C++ e que implementam os modelos de linguagem; as estruturas auxiliares e algumas funções extras; um conjunto de programas executáveis que operam sobre os modelos de linguagem e realizam diversas tarefas; e alguns *scripts* que provêem facilidades de acesso às funcionalidades do *software*. O SRILM é um *software* livre, sendo distribuído sob uma licença muito pouco restritiva. [6]

Metodologia

- ➔ Estudo do software HTK
- ➔ Modelagem Acústica(treinamento do HMM)
- ➔ Modelagem Lingüística(treinamento do modelo de linguagem)
- ➔ Avaliação do desempenho do sistema

Recursos utilizados

Para as etapas do treinamento foram utilizados:

- ✓ Corpora de fala gravado com 90 locutores diferentes, de propriedade da empresa Vocalize, totalizando 8h e 53 min de áudio e 7396 sentenças.
- ✓ Corpora de texto com 7396 sentenças diferentes, mais de 20 mil palavras diferentes no vocabulário.

Atividades desenvolvidas

Para efetuar o treinamento acústico, primeiramente, deve-se obter a transcrição fonética das sentenças e os parâmetros acústicos do aúdio a ser utilizado no treinamento. Os parâmetros dos HMMs são inicialmente, zerados e então, começam a ser estimados, figura 13.

Após isso, ocorre a fusão dos modelos de silêncio e pausa, daí, os parâmetros dos HMMs são reestimados e então, ocorre o realinhamento da transcrição fonética utilizando o HMMs obtidos nas etapas anteriores e ao final, ocorre uma nova reestimação.

Ao final dessa etapa, pode-se realizar o treinamento com monophones com múltiplas gaussianas por estado ou o treinamento com triphones entre palavras múltiplas gaussianas por estado

Foram desenvolvidos scripts em python e outras linguagens de programação para automatizar o processo.

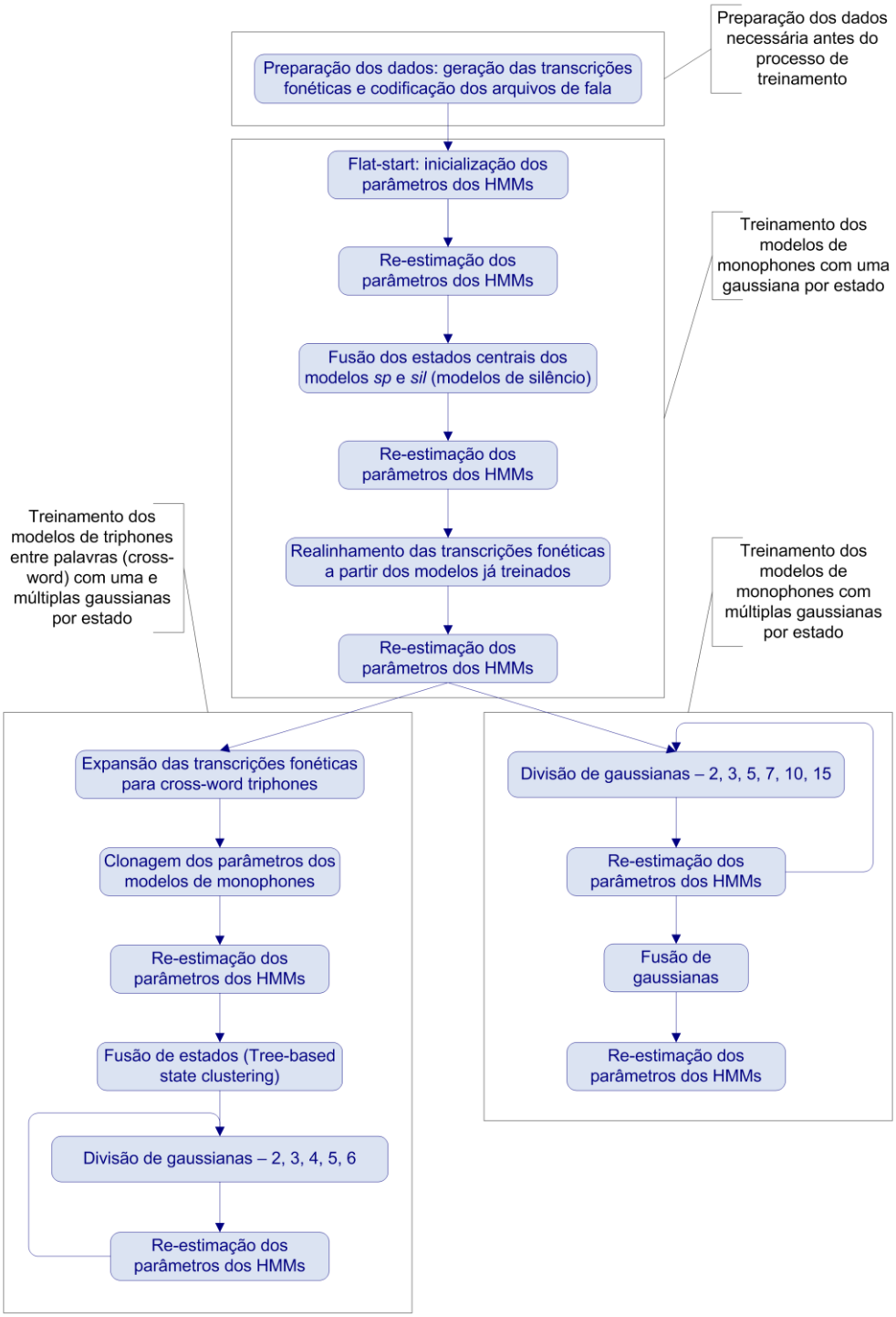


Figura 13 Diagrama de blocos detalhando o funcionamento das etapas de treinamento

Após o treinamento, ocorre o teste com áudios utilizados no treino, permitindo avaliar o nível de precisão do reconhecimento. São realizadas comparações no acerto em palavras e sentenças completas.

Resultados

Experimento realizado no dia 12/05/2010 obteve um acerto de 96.28% das palavras e 74.46% das sentenças, utilizando a base completa.

Utilizando as 237 primeiras sentenças, o resultado demonstrou ser superior, 96.64% das palavras e 77.64% das sentenças.

5. Considerações Finais

O desenvolvimento do trabalho permitiu entender o funcionamento básico de um sistema de reconhecimento de fala, interagindo com areas da computação, engenharia elétrica, linguistica e estatística.

Durante o desenvolvimento do experimento observou-se que o reconhecimento melhora em proporção a quantidade de amostra de audio, sentenças, locutores utilizados no treinamento.

6. Referências bibliográficas

[1] Huang, X., Acero, A., *Spoken Language Processing*. Prentice Hall – PTR, Upper Saddle River, New Jersey, USA, 2001.

[2] Rabiner, L., Juang, B., H., *Fundamentals of Speech Recognition*, Prentice Hall 1993.

[3] Levinson, S., E., *Mathematical Models for Speech Technology*, John Wiley & Sons, 2005.

[4] CHARNIAK, E. Statistical Language Learning. The MIT Press, Massachusetts, 1993.

[5] HTK (2007) *Hidden Markov Model Toolkit V3.4*. Machine Intelligent Laboratory of the Cambridge University, Engineering Department (http://htk.eng.cam.ac.uk/)

[6] SRILM – *The SRI Language Modeling Toolkit* – SRI Speech Technology and Research Laboratory, SRI International (http://www.speech.sri.com/projects/srilm/)